Virtual Machine Trading in a Federation of Clouds: Individual Profit and Social Welfare Maximization

Hongxing Li*, Chuan Wu*, Zongpeng Li† and Francis C.M. Lau*
*Department of Computer Science, The University of Hong Kong, Hong Kong, Email: {hxli, cwu, fcmlau}@cs.hku.hk
†Department of Computer Science, University of Calgary, Canada, Email: zongpeng@ucalgary.ca

Abstract—**By sharing resources among different cloud providers, the paradigm of federated clouds exploits temporal availability of resources and geographical diversity of operational costs for efficient job service. While interoperability issues across different cloud platforms in a cloud federation have been extensively studied, fundamental questions on cloud economics remain: When and how should a cloud trade resources (*e.g.*, virtual machines) with others, such that its net profit is maximized over the long run, while a close-to-optimal social welfare in the entire federation can also be guaranteed? To answer this question, a number of important, inter-related decisions, including job scheduling, server provisioning and resource pricing, should be dynamically and jointly made, while the long-term profit optimality is pursued. In this work, we design efficient algorithms for inter-cloud virtual machine (VM) trading and scheduling in a cloud federation. For VM transactions among clouds, we design a double-auction based mechanism that is strategyproof, individual rational, ex-post budget balanced, and efficient to execute over time. Closely combined with the auction mechanism is a dynamic VM trading and scheduling algorithm, which carefully decides the true valuations of VMs in the auction, optimally schedules stochastic job arrivals with different SLAs onto the VMs, and judiciously turns on and off servers based on the current electricity prices. Through rigorous analysis, we show that each individual cloud, by carrying out the dynamic algorithm in the online double auction, can achieve a time-averaged profit arbitrarily close to the offline optimum. Asymptotic optimality in social welfare is also achieved under homogeneous cloud settings. We carry out simulations to verify the effectiveness of our algorithms, and examine the achievable social welfare under heterogeneous cloud settings, as driven by the real-world Google cluster usage traces.**

I. Introduction

The emerging federated cloud paradigm advocates sharing of disparate cloud services (in separate data centers) from different cloud providers, and interconnects them based on common standards and policies to provide a universal environment for cloud computing. Such a cloud federation exploits temporal and spatial availability of resources (*e.g.*, virtual machines) and diversity of operational costs (*e.g.*, electricity prices): when a cloud experiences a burst of incoming jobs, it may resort to VMs from other clouds with idle resources; when the electricity price for running servers and VMs is high at one cloud data center, the cloud can schedule jobs onto other cloud data centers with lower electricity charge at the moment. In this way, the aggregate job processing capacity of the cloud federation can be potentially higher than the aggregation of capacities of separate clouds operating alone, and the overall profit can be larger.

To implement the federated cloud paradigm, significant interest has arisen on developing interfaces and standards to enable cloud interoperability and job portability across different cloud platforms ([1] [2]). However, fundamental problems on cloud economics remain to be investigated. A cloud in the real world is selfish, and aims to maximize its own profit, *i.e.*, its income from handling jobs and leasing VMs to other clouds subtracting its operational costs and expenses in VM rental from other clouds. Only if its profit can be maximized and in any case not lower than when operating alone, can a cloud be incentivized to join a federation. This calls for an efficient mechanism to carry out resource trading and scheduling among federated clouds, to achieve profit maximization for individual clouds, as well as to perform well in social welfare. A number of inter-related, practical decisions are involved: (1) *VM pricing*: what mechanism should be advocated for VM sale and purchase among the clouds, and at what prices? (2) *Job scheduling*: with time-varying job arrivals at each cloud, targeting different resources and SLA requirements, should a cloud serve the jobs right away or later, to exploit time-varying electricity prices? And should a cloud serve a job using its own resources or others' resources? (3) *Server provisioning*: is it more beneficial for a cloud to keep many of its servers running to serve jobs of its own and from others, or to turn some of them down to save electricity? These decisions should be efficiently and optimally made in an online fashion, while guaranteeing long-term optimality of individual cloud's profits, as well as the social welfare.

In this paper, we design efficient algorithms for inter-cloud resource trading and scheduling, in a federation consisting of disparate cloud data centers. A double-auction based mechanism is proposed for the sell and purchase of available VMs across cloud boundaries over time. The auction is strategyproof, individual rational, ex-post budget balanced, and computationally efficient (polynomial time complexity). Closely combined with the auction mechanism is an efficient, dynamic VM trading and scheduling algorithm, which carefully decides the true valuations of VMs to participate in the auction, optimally schedules randomly-arriving jobs with different resource requirements (*e.g.*, number of VMs) and SLAs (*e.g.*, maximum job scheduling delay) onto different data centers, and judiciously turns on and off servers in the clouds based on the current electricity prices. The dynamic algorithm serves as an efficient strategy for each cloud to employ in the online double auction, and is proven to maximize individual profit for each cloud, over the long run of the system. The contributions of this work are summarized below.

First, among the first in the literature, we address selfishness of individual clouds in a cloud federation, and design efficient mechanisms to maximize the net profit of each cloud. This profit is not only guaranteed to be larger than that when the cloud operates alone, but also maximized over the long run, in the presence of time-varying job arrivals and electricity prices at the cloud.

Second, we novelly combine a truthful double auction mechanism with stochastic Lyapunov optimization techniques, and design an online VM trading and scheduling algorithm, for a cloud to optimally price the VMs and to judiciously schedule the VM and server usages. Each cloud values different VMs based on the back pressure in job queue scheduling, and bids them in the auction for effective VM acquisition.

Third, we demonstrate that by applying the dynamic algorithm in the online double auction, each cloud can achieve a time-averaged profit arbitrarily close to its offline optimum (obtained if the cloud knows complete information on incoming jobs and electricity prices in the entire time span). We also prove that the social welfare, *i.e.*, the time-averaged overall profit in the federation, can be asymptotically maximized when the number of clouds grows, under homogenous cloud settings. Trace-driven simulations examine the achievable social welfare with our dynamic algorithm under heterogenous settings.

In the rest of the paper, we discuss related literature in Sec. II, present the system model in Sec. III, and introduce the detailed resource trading and scheduling mechanisms in Sec. IV. A double auction mechanism is proposed in Sec. V, and a benchmark social-welfare maximization algorithm is discussed in Sec. VI. Theoretical analysis and simulation studies are presented in Sec. VII and Sec. VIII, respectively. Sec. IX concludes the paper.

II. Related work

A. Optimal Scheduling in Cloud Systems

Most existing literature ([3]–[6] and references therein) on resource scheduling in cloud systems focus on a single cloud that operates alone. A common theme is to minimize the operational costs (mainly consisting of electricity bills) in one or multiple data centers of the cloud, while providing certain performance guarantee of job scheduling, *e.g.*, in terms of average job completion times [3]–[6].

Urgaonkar *et al.* [5] propose an algorithm with joint job admission control, routing and resource allocation for power consumption reduction in a virtualized data center. Rao *et al.* [3] advocate minimization of electricity expenses by exploiting the temporal and spatial diversities of electricity prices. Yao *et al.* [6] minimize the power cost with a two-time scale algorithm for delay tolerant workloads. Ren *et al.* [4] also aim to minimize the energy cost while addressing the fairness in resource allocation. All the above works provide *average* delay guarantees for job services.

Different from these studies on a stand-alone cloud with centralized control, this work investigates profit maximization for individual selfish clouds in a federation, where each participant makes its own decisions. Besides, bounded scheduling delay for each job is guaranteed even in worst cases, contrasting the existing solutions that ensure average delays.

B. Resource Trading Mechanisms

A rich body of literature is devoted to resource trading in grid computing [7] and wireless spectrum leasing [8] [9]. Various mechanisms have been studied, *e.g.*, bargaining [7], fixed or dynamic pricing based on a contract or the supply-demand ratio [10], and auctions [8] [9].

A bargaining mechanism [7] typically has an unacceptable complexity by negotiating between each pair of traders. Fixed pricing, *e.g.*, Amazon EC2 on-demand instances, has been shown to be inefficient in social welfare maximization in cases of system dynamics [11]. Dynamic pricing, such as Amazon EC2 spot instances, could be inefficient too, where the participants can quote the resources untruthfully [12].

Auction stands out as a promising mechanism, on which there have been abundant solutions ([8], [9] and references therein) with truthful design and polynomial complexity. Although some recent works [11]–[13] aim to design an auction mechanism with individual rationality (non-negative profit gain) for trading in federated clouds, they do not explicitly address individual profit maximization over the long run, nor other desirable properties such as truthfulness, ex-post budget balance, and social welfare maximization. Moreover, little literature on auctions provides methods to quantitatively calculate the true valuations in each bid, which are simply assumed as known. Our design addresses these issues.

III. System Model and Auction Framework

A. Federation of Clouds

We consider a federation of F clouds, each located at a different geometric location and operates autonomously to gain profit by serving its customers' job requests, managing server provisioning and trading resources with other clouds.

Service demands: Each individual cloud $i \in [1, F]$ has a front-end proxy server, which accepts job requests from its customers. There are S types of jobs serviced at each cloud, each specified by a three-tuple $< m_s, g_s, d_s >$. Here, $m_s \in [1, M]$ specifies the type of the required VM instances, where M is the maximum number of VM types, and each type corresponds to a different set of configurations of CPU, storage and memory; g_s is the number of type-m_s VMs that the job needs simultaneously (See Amazon EC2 API [10]); and d_s stands for the SLA (Service Level Agreement) of job type $s \in [1, S]$, evaluated by the maximal response delay for scheduling a job, *i.e.*, the time-span from when the job arrives to when it starts to run on scheduled VMs. In a cloud in practice, it is common to buy servers of the same configuration and provision the same type of VMs on one machine [14]. Therefore, we suppose each cloud i has N_i^m homogenous servers to provision VMs of type $m \in [1, M]$, each of which can provide a maximum of C_i^m VMs of this type; the total number of servers in cloud i is $\sum_{m=1}^{M} N_i^m$.

The system runs in a time-slotted fashion. At the beginning of each time slot t, $r_i^s(t) \in [0, R_i^s]$ jobs arrive at cloud i, for each job type s. R_i^s is an upper-bound on the number of type-s jobs submitted to cloud i in a time slot. The arrival of jobs is an ergodic process at each cloud. We suppose the arrival rate is given, and how a customer decides which cloud to use is orthogonal to this study. Let $p_i^s(t) \in [0, p_i^{s(max)}]$ be the given service charge to the customer by cloud i, for accepting a job of type s in time slot t, which remains fixed within a time slot, but may vary across time slots. Here, $p_i^{s(max)}$ is the max possible price for $p_i^s(t)$. Such a general charging model subsumes pricing schemes in practice: *e.g.*, time-independent $p_i^s(t)$ corresponds to the *on-demand* VM charging scheme, while time-varying $p_i^s(t)$ can represent the *spot instance* prices based on the current demand *vs.* supply [10].

Job scheduling: Each incoming job to cloud i enters a FIFO queue of its type — a cloud i maintains a queue to buffer unscheduled jobs of each type s, with $Q_i^s(t)$ as its length in t. When the required VMs of a job are allocated, the job departs from its queue and starts to run on the VMs. A cloud may schedule its jobs on either its own VMs or VMs leased from other clouds, for the best economic benefits. Let $\mu_{ij}^s(t)$ be the number of type-s jobs of cloud i that are scheduled for processing in cloud j at the beginning of time slot t.

When a job's demanded maximum response time (the SLA) cannot be met, in cases of system overload, it is dropped. A penalty is enforced in this case, to compensate for the customer's loss. Let

$$D_i^s(t) \in [0, D_i^{s(max)}] \qquad (1)$$

be the number of type-s jobs dropped by cloud i in t, where $D_i^{s(max)}$ is the maximum value of $D_i^s(t)$. Let ξ_i^s be the penalty to drop one such job, which is at least the maximum price charged to customers when accepting the jobs, *i.e.*, $\xi_i^s \geq p_i^{s(max)}$.

Hence, the number of unscheduled jobs buffered at each cloud $i \in [1, F]$ can be updated with the following queueing law:

$$\begin{aligned} Q_i^s(t+1) = &\max\{Q_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^s(t) - D_i^s(t), 0\} \\ &+ r_i^s(t), \quad \forall s \in [1, S]. \end{aligned} \qquad (2)$$

Job scheduling should satisfy the following SLA constraint:

Each type-s job in cloud i is either scheduled or dropped (subject to a penalty) before its maximum response delay d_s, $\forall s \in [1, S]$. (3)

We apply the $\epsilon-$persistence queue technique [15], to create a virtual queue Z_i^s associated with each job queue Q_i^s ($\forall i \in [1, F]$):

$$\begin{aligned} Z_i^s(t+1) = &\max\{Z_i^s(t) + \mathbf{1}_{\{Q_i^s(t)>0\}} \cdot [\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^s(t)] - D_i^s(t) \\ &- \mathbf{1}_{\{Q_i^s(t)=0\}} \cdot \sum_{j=1}^{F} \frac{C_j^{m_s} \cdot N_j^{m_s}}{g_s}, 0\}, \forall s \in [1, S]. \end{aligned} \qquad (4)$$

Here, $\epsilon_s > 0$ is a constant. $\mathbf{1}_{\{Q_i^s(t)>0\}}$ and $\mathbf{1}_{\{Q_i^s(t)=0\}}$ are indicator functions such that

$$\mathbf{1}_{\{Q_i^s(t)>0\}} = \begin{cases} 1 & \text{if } Q_i^s(t) > 0 \\ 0 & \text{Otherwise} \end{cases}; \ \mathbf{1}_{\{Q_i^s(t)=0\}} = \begin{cases} 1 & \text{if } Q_i^s(t) = 0 \\ 0 & \text{Otherwise} \end{cases}.$$

Length of this virtual queue reflects the cumulated response delay of jobs from the respective job queue. Our algorithm seeks to bound the lengths of job queues and virtual queues, with properly set ϵ_s, and hence the maximum response delay of jobs can be bounded, *i.e.*, constraint (3) is satisfied.

Server provisioning: We consider electricity cost, for running and cooling the servers [16], as the main component of the operational cost in a cloud. Other costs, *e.g.*, space rental and labour, remain relatively fixed for a long time, and are of less interest. Given that electricity prices vary at different locations and from time to time [3] [17], we model the operational cost $\beta_i(t)$ in each cloud i as a general ergodic process over time, varying across time slots between $\beta_i^{(min)}$ and $\beta_i^{(max)}$.

Each cloud strategically decides the number of active servers at each time, to optimize its profit. Let $n_i^m(t)$ be the number of active servers provisioning type-m VMs at cloud i in t. The available server capacities at each cloud $i \in [1, F]$ constrain the feasible job scheduling at time t:

$$\sum_{j\in[1,F]} \sum_{s:m_s=m, s\in[1,S]} g_s \mu_{ji}^s(t) \leq C_i^m \cdot n_i^m(t), \ \forall m \in [1, M], \qquad (5)$$

$$n_i^m(t) \leq N_i^m, \ \forall m \in [1, M]. \qquad (6)$$

(5) states that the overall demand for type-m VMs in cloud i from itself and other clouds should be no larger than the maximum number of available type-m VMs on the active servers in cloud i. Here $g_s \mu_{ji}^s(t)$ is the total number of VMs needed by type-s jobs scheduled from cloud j to cloud i in t. Motivated by practical job execution efficiency, we only consider scheduling a job to VMs from a single cloud, but not VMs across different clouds. (6) ensures that the number of active servers is limited by the total number of on-premise servers of the corresponding VM configuration at each cloud.

B. Inter-cloud VM Trading with Double Auction

In an inter-cloud resource market, VMs constitute the items for trading. For each type of VMs, multiple clouds may have them on sale while multiple other clouds can request them. A double auction is a natural fit to implement efficient trading in this case, allowing both selling and buying clouds to actively participate in pricing, on behalf of their own benefits. In our dynamic system, a *multi-unit double auction* is carried out among the clouds at the beginning of each time slot, deciding the VM trades within that time slot.

Buyers & Sellers: A cloud can be both a buyer and a seller. A buy-bid $< b_i^m(t), \gamma_i^m(t) >$ records the unit price and maximum quantity at which cloud i is willing to buy VMs of type m, in t. Similarly, a sell-bid $< s_i^m(t), \eta_i^m(t) >$ records the unit price and maximum quantity at which cloud i is willing to sell VMs of type m in t.

Let $\tilde{b}_i^m(t)$ and $\tilde{s}_i^m(t)$ be cloud i's true valuation of buying and selling a type-m VM respectively (the max/min price it is willing to pay/accept). Similarly, let $\tilde{\gamma}_i^m(t)$ and $\tilde{\eta}_i^m(t)$ be cloud i's true valuation of the quantity to buy and sell VMs of type m respectively (the maximum volume of VMs it is willing to purchase/sell). A cloud i may strategically manipulate the bid prices and volumes, in the hope of maximizing its profit. We show in Sec. VII that the double auction proposed in Sec. IV is truthful, such that each bid price reveals the true valuation.

Auctioneer: We assume that there is a broker in the cloud federation, assuming the role of the auctioneer. After collecting

TABLE I
NOTATION: INPUT QUANTITIES AND INTERMEDIATE VARIABLES

<table>
<tr><td>F</td><td># of clouds</td><td>S</td><td># of service types</td></tr>
<tr><td>M</td><td># of VM types</td><td>m_s</td><td>VM type of service type s</td></tr>
<tr><td>d_s</td><td>Max. response delay of service type s</td><td>g_s</td><td># of VMs required by service type s</td></tr>
<tr><td>$r_i^s(t)$</td><td colspan="3"># of type-s jobs arrived at cloud i, slot t</td></tr>
<tr><td>R_i^s</td><td colspan="3">Max. # of type-s jobs arrived at cloud i per slot</td></tr>
<tr><td>$p_i^s(t)$</td><td colspan="3">Service price for each job of type s at cloud i, slot t</td></tr>
<tr><td>$p_i^{s(max)}$</td><td colspan="3">Max. service price for each type-s job at cloud i per slot</td></tr>
<tr><td>$\beta_i(t)$</td><td colspan="3">Cost for operating an active server at cloud i, slot t</td></tr>
<tr><td>$\beta_i^{(min)}$</td><td colspan="3">Min. cost for operating an active server at cloud i per slot</td></tr>
<tr><td>$\beta_i^{(max)}$</td><td colspan="3">Max. cost for operating an active server at cloud i per slot</td></tr>
<tr><td>ξ_i^s</td><td colspan="3">Penalty for dropping a type-s job at cloud i</td></tr>
<tr><td>$D_i^{s(max)}$</td><td colspan="3">Max. # of type-s jobs cloud i drops per slot</td></tr>
<tr><td>C_i^m</td><td colspan="3">Max. # of type-m VMs an active server at cloud i provisions</td></tr>
<tr><td>N_i^m</td><td colspan="3">Total # of servers provisioning type-m VMs at cloud i</td></tr>
<tr><td>$Q_i^s(t)$</td><td colspan="3">Length of queue buffering type-s jobs at cloud i, slot t</td></tr>
<tr><td>$Z_i^s(t)$</td><td colspan="3">Length of virtual queue of type-s jobs at cloud i, slot t</td></tr>
<tr><td>ϵ_s</td><td colspan="3">Constant positive parameter for $Z_i^s(t), \forall i \in [1, F]$</td></tr>
<tr><td>$Q_i^{s(max)}$</td><td colspan="3">Maximum length of queue $Q_i^s(t)$</td></tr>
<tr><td>$Z_i^{s(max)}$</td><td colspan="3">Maximum length of virtual queue $Z_i^s(t)$</td></tr>
<tr><td>V</td><td colspan="3">User-defined constant positive parameter for dynamic algorithm</td></tr>
</table>

all the buy and sell bids, the auctioneer executes a double auction to be detailed in Sec. V, to decide the set of successful buy and sell bids, their clearing prices and the numbers of VMs to trade in each type. Let $\hat{b}_i^m(t)$ be the actual charge price for cloud i to buy one type-m VM, and $\hat{\gamma}_i^m(t)$ be the actual number of VMs purchased. Similarly, let $\hat{s}_i^m(t)$ be the actual income cloud i receives for selling one type-m VM, and $\hat{\eta}_i^m(t)$ be the actual number of VMs sold.

Let $\alpha_{ij}^m(t)$ be the number of type-m VMs that cloud $i \in [1, F]$ purchases from cloud $j \in [1, F]$ in t, as decided by the auctioneer:

$$\hat{\gamma}_i^m(t) = \sum_{j \in [1,F], j \neq i} \alpha_{ij}^m(t), \ \forall m \in [1, M], \tag{7}$$

$$\hat{\eta}_i^m(t) = \sum_{j \in [1,F], j \neq i} \alpha_{ji}^m(t), \ \forall m \in [1, M]. \tag{8}$$

Since VMs are purchased for serving jobs, the job scheduling decisions $\mu_{ij}^s(t)$ at each cloud $i \in [1, F]$, are related to the number of VMs it purchases:

$$\sum_{s: s \in [1,S], m_s = m} g_s \cdot \mu_{ij}^s(t) = \alpha_{ij}^m(t),$$
$$\forall m \in [1, M], \forall i, j \in [1, F], i \neq j. \tag{9}$$

Three economic properties are desirable for the auctioneer's mechanism. (i) *Truthfulness*: Bidding true valuations is a dominant strategy, and consequently, both bidder strategies and auction design are simplified. (ii) *Individual Rationality*: Each cloud obtains a non-negative profit by participating in the auction. (iii) *Ex-post Budget Balance*: The auctioneer has a non-negative surplus, *i.e.*, the total payment from all winning buy-bids is no less than the total charge for all winning sell-bids in each time slot.

C. Individual Selfishness

Each cloud in the federation aims to maximize its time-averaged profit (revenue minus cost) over the long run of the system, while striking to fulfill the resource and SLA requirements of each job.

TABLE II
NOTATION: DECISION VARIABLES AT INDIVIDUAL CLOUDS

$\mu_{ij}^s(t)$	# of type-s jobs scheduled from cloud i to cloud j, slot t
$n_i^m(t)$	# of active servers providing type-m VMs at cloud i, slot t
$D_i^s(t)$	# of dropped type-s jobs at cloud i, slot t
$\tilde{s}_i^m(t)$	True value of selling one type-m VM from cloud i, slot t
$\tilde{\eta}_i^m(t)$	True value of volume to sell type-m VMs from cloud i, slot t
$s_i^m(t)$	Bid price for selling one type-m VM from cloud i, slot t
$\eta_i^m(t)$	Max. # of type-m VMs cloud i can sell, slot t
$\tilde{b}_i^m(t)$	True value of buying one type-m VM by cloud i, slot t
$\tilde{\gamma}_i^m(t)$	True value of volume to buy type-m VMs by cloud i, slot t
$b_i^m(t)$	Bid price for buying one type-m VM by cloud i, slot t
$\gamma_i^m(t)$	Max. # of type-m VMs cloud i can buy, slot t

TABLE III
NOTATION: DECISION VARIABLES AT THE AUCTIONEER

$\hat{s}_i^m(t)$	Actual price of selling one type-m VM from cloud i, slot t
$\hat{\eta}_i^m(t)$	Actual # of type-m VMs sold from cloud i, slot t
$\hat{b}_i^m(t)$	Actual price of buying one type-m VM by cloud i, slot t
$\hat{\gamma}_i^m(t)$	Actual # of type-m VMs bought by cloud i, slot t
$\alpha_{ij}^m(t)$	Actual # of type-m VMs sold from cloud j to i, slot t
$\theta_j^m(t)$	The j^{th} highest buy-bid price for type-m VMs at auctioneer
$\vartheta_j^m(t)$	The j^{th} lowest sell-bid price for type-m VMs at auctioneer
$L_j^m(t)$	Max. # of type-m VMs to sell, in sell-bid with j^{th} lowest price at auctioneer in t

Revenue: A cloud has two sources of revenue: i) job service charges paid by its customers, and ii) the proceeds from VM sales. The time-averaged revenue of cloud $i \in [1, F]$ by undertaking different types of jobs from its customers is

$$\Phi_1^i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{s \in [1,S]} \mathbb{E}\{p_i^s(t) \cdot r_i^s(t)\}. \tag{10}$$

We assume the front-end charges, $p_i^s(t)$, from a cloud to its customers, are given. Hence, this part of the revenue is fixed in each time slot. The time-averaged income of cloud $i \in [1, F]$ from selling VMs to other clouds is:

$$\Phi_2^i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{m \in [1,M]} \mathbb{E}\{\hat{s}_i^m(t) \cdot \hat{\eta}_i^m(t)\}. \tag{11}$$

Cloud i can control this income by adjusting its sell-bids, *i.e.*, $s_i^m(t)$ and $\eta_i^m(t)$, $\forall m \in [1, M]$, at each time.

Cost: The cost of cloud i consists of three parts: i) operational costs incurred for running its active servers, ii) the penalties for dropping jobs, and iii) the expenditure on buying VMs from other clouds. The time-averaged cost for operating servers at each cloud $i \in [1, F]$ is decided by the number of active servers in each time, *i.e.*,

$$\Psi_1^i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\beta_i(t) \cdot \sum_{m=1}^{M} n_i^m(t)\}. \tag{12}$$

The time-averaged penalty at each cloud $i \in [1, F]$ is determined by the number of dropped jobs over time, *i.e.*, $D_i^s(t), \ \forall s \in [1, S]$, $t \in [0, T-1]$:

$$\Psi_2^i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{s \in [1,S]} \mathbb{E}\{\xi_i^s \cdot D_i^s(t)\}. \tag{13}$$

The time-averaged expenditure for VM purchases is decided by the actual VM trading prices and numbers, as decided by the buy-bids $(b_i^m(t), \gamma_i^m(t))$ from cloud $i \in [1, F]$:

$$\Psi_3^i = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{m=1}^{M} \hat{b}_i^m(t) \cdot \hat{\gamma}_i^m(t)\}. \tag{14}$$

Profit Maximization: The profit maximization problem at cloud $i \in [1, F]$ can be formulated as follows:

$$\begin{aligned} \max \quad & \Phi_1^i + \Phi_2^i - \Psi_1^i - \Psi_2^i - \Psi_3^i \\ s.t. \quad & \text{Constraints (1)-(9).} \end{aligned} \tag{15}$$

D. Social Welfare

Social welfare is the overall profit of the cloud federation:

$$\sum_{i\in[1,F]} (\Phi_1^i + \Phi_2^i - \Psi_1^i - \Psi_2^i - \Psi_3^i).$$

Since the income and expenditure due to VM trades among the clouds cancel each other, the formula above equals $\sum_{i\in[1,F]}(\Phi_1^i - \Psi_1^i - \Psi_2^i)$. The social welfare maximization problem is:

$$\begin{aligned} \max \quad & \sum_{i\in[1,F]} (\Phi_1^i - \Psi_1^i - \Psi_2^i) \\ s.t. \quad & \text{Constraints (1)-(6)}, \ \forall i \in [1, F] \end{aligned} \tag{16}$$

which globally optimizes server provisioning and job scheduling in the federation and maximally serves all the incoming jobs at the minimum cost, regardless of the specific inter-cloud VM trading mechanism.

When a double auction mechanism is truthful, individual rational and ex-post budget balancing, it is shown that efficiency in terms of social welfare maximization cannot be achieved concurrently [18]. We hence make a necessary compromise in social welfare in our auction design, *i.e.*, the sum of maximal individual profits derived by (15) will be smaller than the optimal social welfare from (16). Nevertheless, we will show in Sec. VII and Sec. VIII that our mechanisms still manages to achieve a satisfactory social welfare in the long run.

Tables I, II and III summarize important notation in the paper, for ease of reference.

IV. Dynamic individual-profit maximization algorithm

We next present a dynamic algorithm for each cloud to trade VMs and scheduling jobs/servers, which is in fact applicable under any truthful, individual-rational and ex-post budget balanced double auction mechanism. We will also tailor a double auction mechanism on the auctioneer in the next section. Fig. 1 illustrates the relation among these algorithm modules.

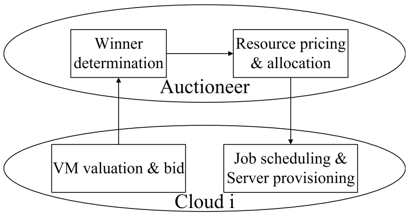

Winner determination
Resource pricing & allocation
Auctioneer
VM valuation & bid
Job scheduling & Server provisioning
Cloud i

Fig. 1. Key algorithm modules.

The goal of the dynamic algorithm at each cloud i is to maximize its time-averaged profit, *i.e.*, to solve optimization (15), by dynamically making decisions in each time slot. We apply the *drift-plus-penalty* framework in Lyapunov optimization theory [19], and derive a one-shot optimization problem to be solved by cloud i in each time slot t as follows. We will prove in Sec. VII that by optimally solving the one-shot optimization at each cloud during each time slot, the dynamic algorithm can achieve a time-averaged individual profit arbitrarily close to its offline optimum (computed with complete knowledge in the entire time span), for each cloud.

A. The One-shot Optimization Problem

Define the set of queues at cloud i in each time slot t as $\Theta_i(t) = \{Q_i^s(t), Z_i^s(t) | s \in [1, S]\}$.

Define the Lyapunov function as follows:

$$L(\Theta_i(t)) = \frac{1}{2} \sum_{s\in[1,S]} [(Q_i^s(t))^2 + (Z_i^s(t))^2].$$

Then the one-slot conditional Lyapunov drift [19] is

$$\Delta(\Theta_i(t)) = L(\Theta_i(t+1)) - L(\Theta_i(t)).$$

Squaring the queuing laws (2) and (4), we can derive the following inequality (details can be found in Appendix A):

$$\begin{aligned} &\Delta(\Theta_i(t)) - V \cdot [\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)] \\ &+ \sum_{s\in[1,S]} [p_i^s(t) \cdot r_i^s(t) - D_i^s(t)\xi_i^s]] \\ \le& B_i + \sum_{s\in[1,S]} [Q_i^s(t)r_i^s(t) + Z_i^s(t)\epsilon_s - Vp_i^s(t) \cdot r_i^s(t)] \\ &- \varphi_1^i(t) - \varphi_2^i(t) - \varphi_3^i(t), \end{aligned} \tag{17}$$

where $V > 0$ is a user-defined positive parameter for gauging the optimality of time-averaged profit, $B_i = \frac{1}{2}\sum_{s\in[1,S]}[(\sum_{j=1}^F C_j^{m_s}N_j^{m_s}/g_s + D_i^{s(max)})^2 + (R_i^s)^2 + (\epsilon_s)^2 + (D_i^{s(max)} + \sum_{j=1}^F C_j^{m_s}N_j^{m_s}/g_s)^2]$ is a constant, and

$$\varphi_1^i(t) = V \sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)],$$

$$\varphi_2^i(t) = \sum_{s=\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^s(t)[Q_i^s(t) + Z_i^s(t)],$$

$$\varphi_3^i(t) = \sum_{s\in[1,S]} D_i^s(t)[Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s].$$

Based on the drift-plus-penalty framework [19], a dynamic algorithm can be derived for each cloud i, which observes the job and virtual queues ($\Theta_i(t)$), job arrival rates ($r_i^s(t), \ \forall s \in [1, S]$), the current cost for server operation ($\beta_i(t)$) in each time slot, and minimizes the RHS of the inequality (17), such that a lower bound for time-averaged profit of cloud i is maximized. Note that $B_i + \sum_{s\in[1,S]}[Q_i^s(t)r_i^s(t) + Z_i^s(t)\epsilon_s - Vp_i^s(t) \cdot r_i^s(t)]$ in the RHS of (17) is fixed in time slot t. Hence, to maximize a lower bound of the time-averaged profit for cloud i, the dynamic algorithm should solve the one-shot optimization problem in each time slot t as follows:

$$\begin{aligned} \max \quad & \varphi_1^i(t) + \varphi_2^i(t) + \varphi_3^i(t) \\ s.t. \quad & \text{Constraints (1), (5)-(9).} \end{aligned} \tag{18}$$

The maximization problem in (18) can be decoupled into two independent optimization problems:

$$\max \quad \varphi_1^i(t) + \varphi_2^i(t) \quad s.t. \ \text{Constraints (5)-(9)}, \tag{19}$$

which is related to optimal decisions on i) buy/sell bids for different types of VMs, and ii) scheduling of active servers and jobs to these servers; and

$$\max \quad \varphi_3^i(t) \quad s.t. \ \text{Constraint (1)}, \tag{20}$$

which is related to optimal decisions on iii) jobs to drop. In the following, we design algorithms to derive the optimal decisions based on problem (19) and problem (20).

B. VM Valuation and Bid

Optimization problem (19) is related to the actual charges that cloud i pays for each type of VMs purchased, $\hat{b}_i^m(t)$ and $\hat{s}_i^m(t)$ ($\forall m \in [1, M]$), and the actual numbers of traded VMs, $\hat{\gamma}_i^m(t)$ and $\hat{\eta}_i^m(t)$ ($\forall m \in [1, M]$), from the double auction. These values are determined by the auctioneer according to buy-bids $(b_i^m(t), \gamma_i^m(t))$ and sell-bids $(s_i^m(t), \eta_i^m(t))$ submitted by all clouds, and its double auction mechanism. That is, each cloud i first proposes its buy-bids and sell-bids to the auctioneer, and then receives the auction results, based on which the *job scheduling and server provisioning* decisions are made. We first investigate how each cloud proposes its buy-bids and sell-bids, and then decide optimal job scheduling and server provisioning in Sec. IV-C.

A truthful double auction is employed at the auctioneer, where sellers and buyers bid their *true values* of the prices and quantities, in order to maximize their individual utilities. (19) is the utility maximization problem for each cloud. If we can find true values of each cloud i, $\tilde{b}_i^m(t), \tilde{\gamma}_i^m(t)$, $\tilde{s}_i^m(t)$ and $\tilde{\eta}_i^m(t)$, and let the cloud bid using these values, the achieved utility in (19) is guaranteed to be the largest, as compared to bidding any other values.

We decide the true values of the bids for each cloud i, according to their definitions in double auctions [8] [9]. The true value of the price to buy (sell) a type-m VM, $\tilde{b}_i^m(t)$ ($\tilde{s}_i^m(t)$), is such a value that, if a VM is purchased (sold) at a price (i) equal to this value, then cloud i's profit remains the same, compared to not obtaining the VM; (ii) higher than this value, a profit loss (gain) at cloud i occurs; and (iii) lower than this value, a profit gain (loss) results. In a multi-unit double auction, the true value of the maximum number of type-m VMs cloud i can buy (sell), $\tilde{\gamma}_i^m(t)$ ($\tilde{\eta}_i^m(t)$), is the maximum number of type-m VMs the cloud is willing to buy (sell) at the true value of the price, *i.e.*, $\tilde{b}_i^m(t)$ ($\tilde{s}_i^m(t)$).

Using the above rationale and based on problem (19), the true values of the buy/sell prices for cloud i can be derived as (detailed derivation steps are given in Appendix B)

$$\tilde{b}_i^m(t) = \frac{Q_i^{s_m^*}(t) + Z_i^{s_m^*}(t)}{V \cdot g_{s_m^*}}, \tag{21}$$

and

$$\tilde{s}_i^m(t) = \begin{cases} \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{V \cdot g_{s_m^*}} & \text{if } \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{V \cdot g_{s_m^*}} > \beta_i(t)/C_i^m \\ \beta_i(t)/C_i^m & \text{Otherwise} \end{cases}, \tag{22}$$

respectively, where

$$s_m^* = arg \max_{s' \in [1,S], m_{s'}=m} \{W_i^{s'}(t)\}, \tag{23}$$

$$\text{and} \qquad W_i^{s'}(t) = \frac{Q_i^{s'}(t) + Z_i^{s'}(t)}{g_{s'}}. \tag{24}$$

Here, $W_i^{s'}(t)$ denotes the weight for scheduling one type-s' job (to run on type-$m_{s'}$ VM(s)) by cloud i in t, and s_m^* specifies the job type with the largest weight (ties broken arbitrarily), among all types of jobs requiring type-m VMs. $W_i^{s'}(t)$ is determined by the following factors: (i) the sum of queue backlogs, $Q_i^{s'}(t) + Z_i^{s'}(t)$, representing the level of urgency for scheduling type-s' jobs in t, since $Q_i^{s'}(t)$ is the number of unscheduled type-s' jobs and $Z_i^{s'}(t)$ reflects the cumulated response delay; (ii) the number of concurrent VMs each type-s' job requires, $g_{s'}$, which decides the job-scheduling difficulty.

The intuition behind (21) and (22) includes: (i) the true value of the price to buy a type-m VM depends on the combined effect of urgency and difficulty for scheduling jobs requiring this type of VMs, and is computed based on the maximum weight that any type of jobs requiring type-m VMs may achieve; (ii) the true value of the price to sell one type-m VM from cloud i is the same as that of the price to buy, if the latter exceeds the current cost of operating a type-m VM in the cloud; otherwise, it is set to the operational cost.

The true values of the number of type-m VMs to buy and to sell at cloud i are

$$\tilde{\gamma}_i^m(t) = \sum_{j \in [1,F]} C_i^m \cdot N_i^m, \tag{25}$$

$$\text{and} \qquad \tilde{\eta}_i^m(t) = C_i^m \cdot N_i^m, \tag{26}$$

respectively. They state that the maximum number of type-m VMs cloud i is willing to buy (sell) at the price in (21) (in (22)), is the number of all potential type-m VMs in the federation. The rationale is as follows: The clearing price for transactions of type-m VMs in the double auction is at most the buyer's true value in (21) and at least the seller's true value in (22), if the corresponding buy/sell bids are successful. By definition of the true value, if the actual charge per VM is lower (higher) than the true value, a profit gain happens at the buyer (seller), and the more VMs purchased (sold), the larger the profit gain. Therefore, a cloud is willing to buy or sell at the largest quantity possible, for profit maximization.[1]

To conclude, in each time slot t, cloud i submits its bids as $b_i^m(t) = \tilde{b}_i^m(t)$, $s_i^m(t) = \tilde{s}_i^m(t)$, $\gamma_i^m(t) = \tilde{\gamma}_i^m(t)$ and $\eta_i^m(t) = \tilde{\eta}_i^m(t)$, for each type of VMs $m \in [1, M]$.

C. Server Provisioning, Job scheduling and Dropping

After receiving results of the double auction (actual charges $\hat{b}_i^m(t)$, $\hat{s}_i^m(t)$, $\forall m \in [1, M]$, and the actual numbers of traded VMs $\hat{\gamma}_i^m(t)$, $\hat{\eta}_i^m(t)$, $\forall m \in [1, M]$, $\alpha_{ji}^{m_s}(t), \forall s \in [1, S], \forall j \in [1, F]$), cloud i schedules its jobs on its local servers and (potentially) purchased VMs from other clouds, decides job drops and the number of active servers to provision, by solving optimization problems (19) and (20).

1) Server provisioning: We start with deriving $n_i^m(t)$, $\forall m \in [1, M]$, by assuming known values of $\hat{s}_i^m(t)$, $\hat{\eta}_i^m(t)$, $\hat{b}_i^m(t)$, $\hat{\gamma}_i^m(t)$, $\alpha_{ij}^m(t)$ and $\mu_{ij}^s(t)$ (we will present the value of $n_i^m(t)$ in terms of these variables). In this case, problem (19) is equivalent to the following minimization problem:

$$\begin{aligned} \min \quad & V\beta_i(t) \sum_{m \in [1,M]} n_i^m(t) \\ s.t. \quad & \text{Constraint (5), (6) and (9).} \end{aligned}$$

[1] It may appear counter-intuitive that a cloud is willing to buy all type-m VMs in the federation, regardless of its number of unscheduled jobs requiring type-m VMs, *i.e.*, $\sum_{s \in [1,S], m_s = m} Q_i^s(t)$. Interestingly, our proof in Sec. VII shows that bidding so in each time slot can achieve a time-averaged profit over the long run that approximates the offline optimum, and our simulation in Sec. VIII shows that it performs better as compared to a bidding strategy that asks for the exact number of VMs to serve the unscheduled jobs.

Since $V\beta_i(t) \geq 0$, the best strategy is to assign the minimal feasible value to $n_i^m(t)$, $\forall m \in [1, M]$, that satisfies constraints (5) and (9), which can be combined into

$$\sum_{s\in[1,S],m_s=m} \mu_{ii}^s(t) \cdot g_s + \sum_{j\neq i} \alpha_{ji}^m(t) \leq C_i^m n_i^m(t).$$

Hence, the optimal number of activated servers at cloud i to provision type-m VM can be calculated as

$$n_i^m(t) = (\sum_{s\in[1,S],m_s=m} \mu_{ii}^s(t) \cdot g_s + \sum_{j\neq i} \alpha_{ji}^m(t))/C_i^m. \qquad (27)$$

These many servers can provide enough type-m VMs for serving local jobs and selling to other clouds.

2) Job scheduling: We now derive $\mu_{ij}^s(t)$, $\forall j \in [1, F]$, $s \in [1, S]$, by assuming known values of $\hat{s}_i^m(t)$, $\hat{\eta}_i^m(t)$, $\hat{b}_i^m(t)$, $\hat{\gamma}_i^m(t)$ and $\alpha_{ij}^m(t)$, with $n_i^m(t)$ given in Eqn. (27). Problem (19) is equivalent to the following maximization problem:

$$\begin{aligned} \max \quad & \sum_{s\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^s(t)[Q_i^s(t) + Z_i^s(t)] \\ & - V\beta_i \sum_{s\in[1,S],m_s=m} \mu_{ii}^s(t) \cdot \frac{g_s}{C_i^m} \\ s.t. \quad & \text{Constraint (5), (6) and (9).} \end{aligned}$$

This is a maximum-weight scheduling problem, with $Q_i^s(t)+Z_i^s(t)$ as the per-job scheduling weight for each $\mu_{ij}^s(t)$ ($j \neq i$) and $Q_i^s(t)+Z_i^s(t)-\frac{V\beta_i(t)g_s}{C_i^m}$ as the per-job scheduling weight for each $\mu_{ii}^s(t)$. There are two cases:

$\triangleright$ $j = i$: In this case, by combining constraints (5), (6) and (9), we have

$$\sum_{s:m_s=m,s\in[1,S]} g_s\mu_{ii}^s(t) \leq C_i^m N_i^m - \sum_{j\neq i} \alpha_{ji}^m(t).$$

Based on the above maximum-weight problem, we know that the best strategy is to assign all the remaining type-m_s VMs in cloud i, $C_i^{m_s} n_i^{m_s}(t) - \sum_{j\neq i} \alpha_{ji}^{m_s}(t)$ (the maximum number of on-premise type-m_s VMs minus those sold to other clouds), to serve its own jobs of service type $s^*_{m_s}$ with the largest per-VM scheduling weight $\frac{Q_i^s(t)+Z_i^s(t)}{g_s} - \frac{V\beta_i(t)}{C_i^{m_s}}$ if it is positive (equivalently, the largest $\frac{Q_i^s(t)+Z_i^s(t)}{g_s}$ if $\frac{Q_i^s(t)+Z_i^s(t)}{g_s} > \frac{V\beta_i(t)}{C_i^{m_s}}$), among all job types requiring type-m_s VMs. Otherwise, cloud i does not serve any jobs using its own servers in t. Hence, we derive the optimal number of cloud i's type-s jobs scheduled to run on the cloud's local servers as

$$\mu_{ii}^s(t) = \begin{cases} \frac{C_i^{m_s} \cdot N_i^{m_s} - \sum_{j\neq i} \alpha_{ji}^{m_s}(t)}{g_s} & \text{if } \frac{Q_i^s(t)+Z_i^s(t)}{\cdot g_s} > \frac{V\beta_i(t)}{C_i^{m_s}} \\ & \text{and } \; s = s^*_{m_s} \\ 0 & \text{Otherwise} \end{cases}. \qquad (28)$$

$\triangleright$ $j \neq i$: $\mu_{ij}^s(t)$ can be directly derived by $\alpha_{ij}^{m_s}(t)$, which is the number of type-m_s VMs cloud i purchased from cloud j (constraint (5) is satisfied by our server provisioning decision in Eqn. (27), and constraint (6) is met by Eqn. (28) and (27)), based on constraint (9). Similar to the previous case, we know that the best strategy is to assign all the type-m_s VMs purchased, $\alpha_{ij}^{m_s}(t)$, to serve jobs of service type $s^*_{m_s}$ with the largest per-VM scheduling weight $\frac{Q_i^s(t)+Z_i^s(t)}{g_s}$, as defined in Eqn. (23) and (24). Hence, we derive the optimal solution to the number of type-s jobs to run at cloud $j (\neq i)$ as

$$\mu_{ij}^s(t) = \begin{cases} \alpha_{ij}^{m_s}(t)/g_s & \text{if } s = s^*_{m_s} \\ 0 & \text{Otherwise} \end{cases}. \qquad (29)$$

3) Job dropping: Problem (20) is a maximum-weight problem with weight $Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s$ for job-dropping decision variable $D_i^s(t)$, $\forall s \in [1, S]$, in the objective function. If the weight $Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s > 0$ (*i.e.*, if the level of urgency for scheduling type-s jobs $Q_i^s(t) + Z_i^s(t)$ exceeds the weighted job-drop penalty $V \cdot \xi_i^s$), type-s jobs in queue Q_i^s should be dropped at the maximum rate, *i.e.*, $D_i^s(t) = D_i^{s(max)}$, in order to maximize the objective function value; otherwise, there is no drop, *i.e.*, $D_i^s(t) = 0$. Therefore, the optimal number of type-s jobs dropped by cloud i in t is

$$D_i^s(t) = \begin{cases} D_i^{s(max)} & \text{if } Q_i^s(t) + Z_i^s(t) > V \cdot \xi_i^s \\ 0 & \text{Otherwise} \end{cases}. \qquad (30)$$

In the above results, we note that the derived job scheduling and drop numbers do not need to be bounded by the number of unscheduled jobs in the corresponding job queue, *i.e.*, $\mu_{ij}^s(t)$ and $D_i^s(t)$ are not required to be bounded by $Q_i^s(t)$ according to Eqn. (2). Nevertheless, the actual number of jobs to schedule/drop when running the algorithm, is upper bounded by the length of the job queue.

D. The Dynamic Algorithm

Alg. 1 summarizes the dynamic algorithm for each cloud to carry out in each time slot, in order to maximize its time-averaged profit over the long run.

Algorithm 1 Dynamic Profit Maximization Algorithm at cloud i in Time Slot t

```
Input: $r_i^s(t)$, $Q_i^s(t)$, $Z_i^s(t)$, $g_s$, $m_s$, $\xi_i^s$, $C_i^m$, $N_i^m$ and $\beta_i(t)$, $\forall s \in [1, S]$.
Output: $b_i^m(t)$, $s_i^m(t)$, $\gamma_i^m(t)$, $\eta_i^m(t)$, $D_i^s(t)$, $\mu_{ij}^s(t)$ and $n_i^m(t)$, $\forall m \in [1, M], s \in [1, S], j \in [1, F]$.
1: VM valuation and bid: Decide $b_i^m(t)$, $s_i^m(t)$, $\gamma_i^m(t)$ and $\eta_i^m(t)$ with Eqn. (21)-(26);
2: Server provisioning, job scheduling and dropping: Decide $\mu_{ij}^s(t)$, $D_i^s(t)$ and $n_i^m(t)$ with Eqn. (29), (28), (30) and (27);
3: Update $Q_i^s(t)$ and $Z_i^s(t)$ with Eqn. (2) and (4).
```

We analyze the computation and communication complexities of Alg. 1 as follows.

Computation complexity: We study the computation complexity for each algorithm module respectively.

$\triangleright$ *VM valuation and bid*: The algorithm should first calculate the value of s^*_m for each VM type $m \in [1, M]$ with Eqn. (23) by comparing the weights $W_i^{s'}(t)$ among different types of jobs. In fact, the weight for each job type $s \in [1, S]$ is only evaluated once since it is only involved in the calculation of s^*_m where $m = m_s$. Hence, the computation overhead to find s^*_m, $\forall m \in [1, M]$, is $O(S)$. Based on the value of s^*_m, the buy/sell bids of type-m VMs can be decided by Eqn. (21)-(26) in constant time. For all M VM types, the computation overhead is $O(M)$. Hence, the overall computation complexity for this algorithm module is $O(S + M)$.

$\triangleright$ *Server provisioning, job scheduling and dropping*: With s^*_m, $\forall m \in [1, M]$, calculated in the above algorithm module, we can directly know the value of $s^*_{m_s}$, $\forall s \in [1, S]$. Then, the job scheduling decision $\mu_{ij}^s(t)$ for job type s can be made in constant time based on Eqn. (29) and (28). For all S job types, the computation overhead is $O(S)$.

The server provisioning decisions can be found in constant time based on the job scheduling decisions and the auction results, according to Eqn. (27) for type-m VMs. For all M VM types, the computation overhead is $O(M)$.

Job dropping is also decided in constant time for type-s jobs based on Eqn. (30). For all S job types, the computation complexity is $O(S)$.

$\triangleright$ *Queue update*: For each job type s, the job queue $Q_i^s(t)$ and virtual queue $Z_i^s(t)$ can be updated in constant time based on Eqn. (2) and (4). Hence, for all S job types, the computation overhead is $O(S)$.

In summary, the computation complexity of Alg. 1 is $O(S+M)$.

Communication complexity: The input to Alg. 1 is mostly derived from local information. There is no direct information exchange among individual clouds. The only communication overhead is incurred when a cloud sends its VM bids to the auctioneer and receives the auction results for each VM type. Since there are M VM types, the communication complexity is $O(M)$ for each cloud.

V. Double Auction Mechanism

We next design a double auction mechanism for inter-cloud VM trading, which not only is truthful, individual rational and ex-post budget balanced, but also can enable satisfactory social welfare (Theorems 2-4 and 8, Sec. VII).

The true values of buy and sell bids at each participating cloud (Eqn. (21)-(26)) are not related to the detailed auction mechanism. The true values of the maximum numbers of VMs a cloud is willing to trade ($\hat{\gamma}_i^m(t)$ and $\hat{\eta}_i^m(t)$ in (25) and (26)) are time-independent constants determined by system parameters C_i^m and N_i^m. These parameters, and thus $\hat{\gamma}_i^m(t)$ and $\hat{\eta}_i^m(t)$, are easily known to other clouds, and hence it is not meaningful for a buyer/seller to bid otherwise. We correspondingly design a double auction where $\gamma_i^m(t)$ in each buy-bid is fixed to the value in (25) and $\eta_i^m(t)$ in each sell-bid is always the value in (26), while the buy/sell prices, $b_i^m(t)$'s and $s_i^m(t)$'s, can be decided by the respective buyers/sellers.

The following mechanism is carried out by the auctioneer at the beginning of each time slot t, to decide the actual trading price and number for each type of VMs $m \in [1, M]$.

1. Winner Determination: The auctioneer sorts all received buy-bids for type-m VMs in descending order in the buy prices. Let $\theta_j^m(t)$ be the j^{th} highest. Two buy-bids with the *largest* and *second largest* prices, $\theta_1^m(t)$, $\theta_2^m(t)$, are identified (ties broken arbitrarily). The sell-bids for type-m VMs are sorted in ascending order in the sell prices. Let $\vartheta_j^m(t)$ be the j^{th} lowest, with $L_j^m(t)$ as the corresponding maximum number of VMs to sell, such that $\vartheta_1^m(t) \le \vartheta_2^m(t) \le \ldots \le \vartheta_N^m(t)$. Let j' be the critical index in the sorted sequence of sell-bids, such that $\vartheta_{j'}^m(t)$ is the largest sell price not exceeding $\theta_2^m(t)$, *i.e.*,

$$\vartheta_{j'}^m(t) \le \theta_2^m(t), \text{ and } \vartheta_{j'+1}^m(t) > \theta_2^m(t). \tag{31}$$

If there are at least two sell-bids $\vartheta_1^m(t)$ and $\vartheta_2^m(t)$ no higher than the second largest buy price $\theta_2^m(t)$, the highest buy-bid $\theta_1^m(t)$ wins, and the sell-bids with the lowest to the $(j'-1)^{\text{th}}$ lowest sell prices ($\vartheta_j^m(t) \le \vartheta_{j'}^m(t)$, not including j') win. Otherwise, no buy/sell bid wins.

2. Pricing and Allocation: It is a NP-hard problem to clear the double auction market with discriminatory prices [20]. We apply a uniform clearing price to winning buy/sell bids of type-m VMs, as follows.

$\triangleright$ The price charged to each buyer cloud i of type-m VMs is

$$\hat{b}_i^m(t) = \begin{cases} \theta_2^m(t) & \text{if bid } b_i^m(t) \text{ wins,} \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

$\triangleright$ The price paid to each seller cloud i of type-m VMs is

$$\hat{s}_i^m(t) = \begin{cases} \vartheta_{j'}^m(t) & \text{if bid } s_i^m(t) \text{ wins,} \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

$\triangleright$ The number of type-m VMs bought by cloud i is

$$\hat{\gamma}_i^m(t) = \begin{cases} \sum_{j=1}^{j'-1} L_j^m(t) & \text{if bid } b_i^m(t) \text{ wins,} \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

$\triangleright$ The number of type-m VMs sold by cloud i is

$$\hat{\eta}_i^m(t) = \begin{cases} \eta_i^m(t) & \text{if bid } s_i^m(t) \text{ wins,} \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

$\triangleright$ The number of type-m VMs sold from cloud j to cloud i is

$$\alpha_{ij}^m(t) = \begin{cases} \eta_j^m(t) & \text{if bids } b_i^m(t) \text{ and } s_j^m(t) \text{ win,} \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

For example, consider a federation of 4 clouds with buy and sell prices bid in Table IV, each seeking to buy/sell one VM. Clouds 2 and 3 bid the two largest buy prices \$20 and \$15, which are higher than sell prices from clouds 1 and 4. Hence the buyer cloud 2 and the seller cloud 4 win, while the clearing buy and sell prices are \$15 and \$13, respectively.

TABLE IV
Double auction bids: an illustrative example

	Cloud 1	Cloud 2	Cloud 3	Cloud 4
Buy-bid	\$10	\$20	\$15	\$8
Sell-bid	\$13	\$22	\$16	\$9

VI. Dynamic Social-Welfare Maximization Algorithm: a Benchmark

We also present a dynamic algorithm that maximizes the time-averaged social welfare in the federation (optimization problem (16)), and its derivation steps based on the Lyapunov optimization framework. This algorithm is used as a benchmark to examine the efficiency of Alg. 1 in social welfare.

A. Derivation Details

Similar to the derivation of Alg. 1, we first derive a one-shot optimization problem (40) for the federation to solve based on the *drift-plus-penalty* framework of Lyapunov optimization, and then derive the dynamic benchmark algorithm to solve it optimally in each time slot.

In each time slot t, define the set of queues $\Theta(t)$ in the federation as

$$\Theta(t) = \{Q_i^s(t), Z_i^s(t) | i \in [1, F], s \in [1, S]\}.$$

Define the Lyapunov function as follows:

$$L(\Theta(t)) = \frac{1}{2} \sum_{i \in [1,F]} \sum_{s \in [1,S]} [(Q_i^s(t))^2 + (Z_i^s(t))^2].$$

Then the one-slot conditional Lyapunov drift is

$$\Delta(\Theta(t)) = L(\Theta(t+1)) - L(\Theta(t)).$$

Squaring the queuing laws Eqn. (2) and (4), we can derive the following inequality (details can be found in Appendix C)

$$\begin{aligned}&\Delta(\Theta(t)) + V \cdot \sum_{i\in[1,F]} [\sum_{m\in[1,M]} [\beta_i(t) n_i^m(t)] + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s \\ &- \sum_{s\in[1,S]} p_i^s(t) r_i^s(t)] \\ \le& B + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t) r_i^s(t) + Z_i^s(t)\epsilon_s - V p_i^s(t) r_i^s(t)] \\ &- \varphi_1(t) - \varphi_2(t), \qquad (37)\end{aligned}$$

where $V > 0$ is a user-defined positive parameter for gauging the optimality of the time-averaged social welfare, $B = \sum_{i\in[1,F]} B_i$ is a constant with $B_i = \frac{1}{2}\sum_{s\in[1,S]}[[\sum_{j=1}^F C_j^{m_s} N_j^{m_s}/g_s + D_i^{s(max)}]^2 + [R_i^s]^2 + [\epsilon_s]^2 + [D_i^{s(max)} + \sum_{j=1}^F C_j^{m_s} N_j^{m_s}/g_s]^2]$, and

$$\begin{aligned}\varphi_1(t) = \sum_{i\in[1,F]} [\sum_{s\in[1,S]} [Q_i^s(t) + Z_i^s(t)] \cdot \sum_{j\in[1,F]} \mu_{ij}^s(t) \\ - V\beta_i(t) \sum_{m\in[1,M]} n_i^m(t)], \qquad (38)\end{aligned}$$

$$\varphi_2(t) = \sum_{i\in[1,F]} \sum_{s\in[1,S]} D_i^s(t)[Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s]. \qquad (39)$$

Based on the drift-plus-penalty framework [19], a dynamic algorithm can be derived for the federation to observe job and virtual queues $\Theta(t)$, job arrival rates ($r_i^s(t)$, $\forall i \in [1,F], s \in [1,S]$), the current cost for server operation ($\beta_i(t)$, $\forall i \in [1,F]$) in each time slot, and minimizes the RHS of the inequality (37), such that a lower bound for the time-averaged social welfare is maximized. Note that $B + \sum_{i\in[1,F]}\sum_{s\in[1,S]}[Q_i^s(t) r_i^s(t) + Z_i^s(t)\epsilon_s - V p_i^s(t) r_i^s(t)]$ in the RHS of (37) is fixed in time slot t. Hence, to maximize a lower bound of the time-averaged social welfare for the federation, the dynamic algorithm should solve the one-shot optimization problem in each time slot t as follows:

$$\begin{aligned}\max \quad & \varphi_1(t) + \varphi_2(t) \qquad (40)\\ s.t. \quad & \text{Constraints (1), (5)-(6)}, \ \forall i \in [1,F].\end{aligned}$$

The maximization problem in (40) can be decoupled into two independent optimization problems:

$$\max \ \varphi_1(t) \quad s.t. \ \text{Constraint (5)-(6)}, \ \forall i \in [1,F], \qquad (41)$$

which is related to decisions on job scheduling and server provisioning, and

$$\max \ \varphi_2(t) \quad s.t. \quad \text{Constraint (1)}, \ \forall i \in [1,F], \qquad (42)$$

which is related to decisions on job dropping. We note that to maximize social welfare, the decisions that the federation needs to make are not related to any inter-cloud VM trading mechanism, since the income and expenditure due to VM trades among the clouds have canceled each other when calculating the social welfare. We next solve problem (41) and problem (42) to derive the optimal decisions.

1) Server provisioning: We start with solving $n_i^m(t)$, $\forall m \in [1,M]$, $i \in [1,F]$, by assuming known values of job scheduling decisions $\mu_{ij}^s(t)$'s, and present the value of the former in terms of the latter. In this case, problem (41) is equivalent to the following minimization problem:

$$\begin{aligned}\min \quad & V \sum_{i\in[1,F]} \beta_i(t) \sum_{m\in[1,M]} n_i^m(t) \\ s.t. \quad & \text{Constraint (5) - (6)}, \ \forall i \in [1,F].\end{aligned}$$

Since $V\beta_i(t) \ge 0$, the best strategy is to assign the minimal feasible value to $n_i^m(t)$, for each VM type m at each cloud i, that satisfies constraints (5) and (6). Hence, the optimal number of activated servers at cloud i to provision type-m VM is

$$n_i^m(t) = [\sum_{j\in[1,F]} \sum_{s\in[1,S],m_s=m} \mu_{ji}^s(t) \cdot g_s]/C_i^m. \qquad (43)$$

2) Job scheduling: We next derive $\mu_{ij}^s(t)$, $\forall i \in [1,F]$, $j \in [1,F]$, $s \in [1,S]$, with $n_i^m(t)$ given in Eqn. (43). Problem (41) is equivalent to the following maximization problem:

$$\begin{aligned}\max \quad & \sum_{i\in[1,F]} \sum_{s\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^s(t) \cdot [Q_i^s(t) + Z_i^s(t) - V\beta_j \cdot \frac{g_s}{C_j^{m_s}}] \\ s.t. \quad & \text{Constraint (5) - (6)}, \ \forall i \in [1,F].\end{aligned}$$

This is a maximum-weight scheduling problem, with $Q_i^s(t) + Z_i^s(t) - \frac{V\beta_j(t) g_s}{C_j^{m_s}}$ as the per-job scheduling weight for each $\mu_{ij}^s(t)$. Combining constraints (5) and (6), we have

$$\sum_{i\in[1,F]} \sum_{s:m_s=m,s\in[1,S]} g_s \mu_{ij}^s(t) \le C_j^m N_j^m, \ \forall j \in [1,F].$$

The best strategy is to assign all the type-m VMs in cloud j at the number of $C_j^m N_j^m$ to serve jobs of type $\acute{s}_m$ of cloud $\acute{i}_m$ with the maximum per-VM scheduling weight $\frac{Q_i^s(t)+Z_i^s(t)}{g_s} - \frac{V\beta_j(t)}{C_j^{m_s}}$ if it is positive (equivalently, the largest $\frac{Q_i^s(t)+Z_i^s(t)}{g_s}$ as defined in Eqn. (24) and (23) in Sec. IV if $\frac{Q_i^s(t)+Z_i^s(t)}{g_s} > \frac{V\beta_j(t)}{C_j^{m_s}}$), among all job types from all clouds requiring type-m VMs. Hence, the optimal solution to the number of type-s jobs of cloud i to run at cloud j is

$$\mu_{ij}^s(t) = \begin{cases} C_j^{m_s} \cdot N_j^{m_s}/g_s & \text{if } \frac{Q_i^s(t)+Z_i^s(t)}{g_s} > \frac{V\beta_i(t)}{C_j^{m_s}} \\ & \text{and } \ < i, s >=< \acute{i}_m, \acute{s}_m >, \\ 0 & \text{Otherwise,} \end{cases} \qquad (44)$$

where

$$< \acute{i}_m, \acute{s}_m >= arg \max_{i\in[1,F],s\in[1,S],m_s=m} \{W_i^s(t)\}, \qquad (45)$$

and $W_i^s(t)$ is the weight defined in Eqn. (24).

3) Job dropping: Problem (42) is a maximum-weight problem with weight $Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s$ for job-dropping decision $D_i^s(t)$ in the objective function. If $Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s > 0$, type-$s$ jobs at cloud i should be dropped at the maximum rate; otherwise, there is no drop. Hence, the optimal number of type-s jobs dropped by cloud i in t is

$$D_i^s(t) = \begin{cases} D_s^{(max)} & \text{if } Q_i^s(t) + Z_i^s(t) > V \cdot \xi_i^s \\ 0 & \text{Otherwise.} \end{cases} \qquad (46)$$

B. The Dynamic Benchmark Algorithm

Alg. 2 summarizes the dynamic algorithm for the federation to carry out (*e.g.*, on a centralized controller) in each time slot, in order to maximize its time-averaged social welfare over the long run.

Algorithm 2 Dynamic Social Welfare Maximization Algorithm in Time Slot t

```
**Input**: $r_i^s(t)$, $Q_i^s(t)$, $Z_i^s(t)$, $g_s$, $m_s$, $\xi_i^s$, $C_i^m$, $N_i^m$ and $\beta_i(t)$, $\forall i \in [1,F], s \in [1,S]$.
**Output**: $D_i^s(t)$, $\mu_{ij}^s(t)$ and $n_i^m(t)$, $\forall i \in [1,F], m \in [1,M], s \in [1,S]$.
**Job scheduling and server provisioning**: Decide $\mu_{ij}^s(t)$ and $n_i^m(t)$ with Eqn. (44) and (43);
**Job dropping**: Decide $D_i^s(t)$ with Eqn. (46);
Update $Q_i^s(t)$ and $Z_i^s(t)$ with Eqn. (2) and (4).
```

VII. Performance Analysis

We next analyze the performance guarantee provided by our dynamic individual-profit maximization algorithm and the double auction mechanism.

A. Properties of the Double Auction Mechanism

Theorem 1 (True Valuation): The VM valuations on buy-bids, *i.e.*, Eqn. (21) and (25), and sell-bids, *i.e.*, Eqn. (22) and (26), are true values.

This theorem is proved based on the definition of the true values and the optimization problem (18) solved in each time slot by each cloud in Appendix B.

Theorem 2 (Truthfulness): Bidding truthfully is the dominant strategy of each cloud in the double auction in Sec. V, *i.e.*, no cloud can achieve a higher profit in (18) by bidding with values other than its true values of the buy and sell bids, in Eqn. (21)(25)(22)(26).

We prove this theorem by contradiction and show that, in all cases, no cloud can do better with problem (18) by bidding untruthfully. Details are in Appendix D.

Theorem 3 (Individual Rationality): No winning buyer pays more than its buy-bid price, and no winning seller is paid less than its sell-bid price, *i.e.*, $\hat{b}_i^m(t) \le b_i^m(t)$ and $\hat{s}_i^m(t) \ge s_i^m(t),\ \forall i \in [1,F], m \in [1,M]$.

This theorem can be proved based on the winner determination and pricing schemes in our auction mechanism, with details in Appendix E. Given that the buy-bid (sell-bid) price is the true value of the buyer (seller), this theorem implies that a cloud can receive a non-negative profit gain, if it successfully sells or buys VMs. Hence, *a cloud's profit obtained in a federation with potential VM trades with others, is always no lower than that obtained when operating alone.*

Theorem 4 (Ex-post Budget Balance): At the auctioneer, the total payment collected from the buyers is no smaller than the overall price paid to the sellers, *i.e.*,

$$\sum_{i\in[1,F]} [\hat{b}_i^m(t) \cdot \hat{\gamma}_i^m(t) - \hat{s}_i^m(t) \cdot \hat{\eta}_i^m(t)] \ge 0,\ \forall m \in [1,M].$$

This theorem is proved based on Eqn. (31) - (33), with details in Appendix F.

B. SLA Guarantee

Lemma 1: Let $Q_i^{s(max)} = V\xi_i^s + R_i^s$ and $Z_i^{s(max)} = V\xi_i^s + \epsilon_s$. If $D_i^{s(max)} \ge \max\{R_i^s, \epsilon_s\}$, each job queue $Q_i^s(t)$ and each virtual queue $Z_i^s(t)$ are upper-bounded by $Q_i^{s(max)}$ and $Z_i^{s(max)}$, respectively, in $t \in [0, T-1]$, $\forall i \in [1,F], s \in [1,S]$.

This lemma can be proved by analyzing the job drop decision in (30) and the queue updates in (2)(4). The condition $D_i^{s(max)} \ge \max\{R_i^s, \epsilon_s\}$ ensures that, when the queue lengths grow to satisfy the job drop condition, any further increase on the queues, *e.g.*, R_i^s and ϵ_s, can be balanced by dropping enough number of jobs at the rate of $D_i^{s(max)}$. Detailed proof is included in Appendix G.

Theorem 5 (SLA Guarantee): Each job of type $s \in [1,S]$ is either scheduled or dropped with Alg. 1 before its maximum response delay d_s, if we set $\epsilon_s = \frac{Q_i^{s(max)} + Z_i^{s(max)}}{d_s}$.

This theorem can be proved based on Lemma 1 and the ϵ-persistence queue techniques [15]. The condition on ϵ_s is to ensure that the queue lengths can grow to satisfy the job drop condition, *i.e.*, $Q_i^s + Z_i^s(t) > V\xi_i^s$, if some jobs remain unscheduled in the last d_s slots. Note that a cloud only drops jobs strategically, to balance the loss due to the job drop penalties and the gain in saving VMs for other jobs. For more details, please refer to Appendix H.

C. Optimality of Individual Profit and Social Welfare

Theorem 6 (Individual Profit Optimality): Let Ω_i^* be the offline optimum of time-averaged profit of cloud $i \in [1,F]$, obtained in a truthful, individual-rational, ex-post budget-balanced double auction, with complete information on its own job arrivals and prices in the entire time span $[0, T-1]$. The dynamic Algorithm 1 can achieve a time-averaged profit Ω_i for cloud i within a constant gap B_i/V to Ω_i^*, *i.e.*,

$$\Omega_i \ge \Omega_i^* - B_i/V,$$

where $V > 0$ and $B_i = \frac{1}{2}\sum_{s\in[1,S]}[[\sum_{j=1}^F C_j^{m_s} N_j^{m_s}/g_s + D_i^{s(max)}]^2 + [R_i^s]^2 + [\epsilon_s]^2 + [D_i^{s(max)} + \sum_{j=1}^F C_j^{m_s} N_j^{m_s}/g_s]^2]$ is a constant.

The proof to this theorem is rooted in the Lyapunov optimization theory [19]. The gap B_i/V can be close to zero by fixing ϵ_s and increasing V. Detailed proof is included in Appendix I.

Theorem 7 (Social Welfare Optimality of Alg. 2): Let Π^* be the offline optimum of the time-averaged social welfare in (16), obtained with full information of the federation over the entire time span $[0, T-1]$. The time-averaged social welfare achieved by all clouds by running Alg. 2, approaches the offline-optimal social welfare Π^*, by a constant gap B/V, *i.e.*,

$$\Pi \ge \Pi^* - B/V,$$

where $V > 0$ and $B = \sum_{i\in[1,F]} B_i$. B_i is defined in Theorem 6, $\forall i \in [1,F]$.

The proof to this theorem is also based on the Lyapunov optimization theory [19]. The gap B/V can be close to zero by fixing ϵ_s and increasing V. Detailed proof is included in Appendix J.

Theorem 8 (Asymptotic Optimality in Social Welfare of Alg. 1): Let Π^* be the offline optimum of the time-averaged social welfare in (16), obtained with full information of the federation over the entire time span $[0, T-1]$. Suppose all clouds are homogenous, *i.e.*, with the same number of servers (N_i^m) and the same maximum per-server VM provisioning (C_i^m) for each VM type m, with i.i.d. service prices, job arrivals and operational costs. When the number of clouds, F, grows, the sum of time-averaged profits achieved by all clouds by running Alg. 1 under the double auction mechanism in Sec. V, approaches the offline-optimal social welfare Π^*, by a constant gap B/V, *i.e.*,

$$\Pi \geq \Pi^* - B/V,$$

where $V > 0$ and $B = \sum_{i \in [1,F]} B_i$. B_i is defined in Theorem 6, $\forall i \in [1, F]$.

To prove the theorem, we demonstrate that, when the number of clouds goes to infinity, the one-shot social welfare obtained with Alg. 1 is the same as that achieved by the dynamic benchmark Alg. 2 in the same time slot. Details are in Appendix K.

VIII. Performance Evaluation

A. Simulation Setup

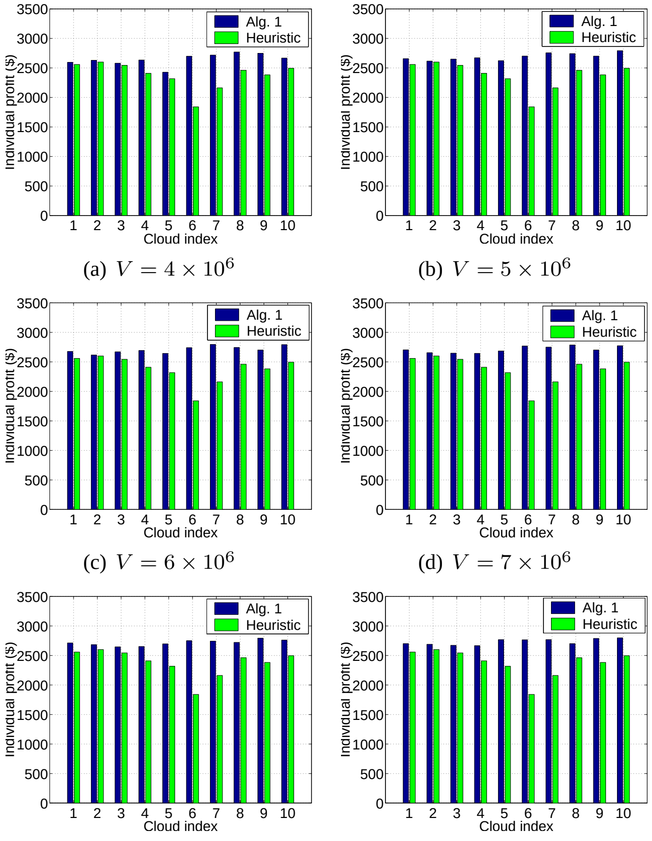
We carry out trace-driven simulation studies based on Google cluster-usage data [21] [22], which record jobs submitted to the Google cluster, with information on their resource demands (CPU, RAM, etc.) and relative charges. We translate the data into concrete job arrival rates, resource types and prices, to drive our simulations as follows.

We consider 24 types of jobs ($< m_s, g_s, d_s >$), 6 VM types (m_s) combined from {*small, median, large*} *CPU* and {*small, large*} *Memory*, and two *SLA* levels (d_s), corresponding to a larger maximum respond delay and a smaller maximum response delay at half of the former. Each job requires either 1 VM or 2 VMs concurrently (g_s).

There are 10 clouds in the federation. One time slot is 1 hour. The number of servers in each cloud that provision VMs of each type ranges within $[800, 1000]$. Each server can provide 30 *small-memory* VMs or 10 *large-memory* VMs. The VM charge to the customer is decided by multiplying g_s by the relative VM price in the Google data, and then by the unit VM price in the range of [0.05, 0.08] $/h. The penalty for dropping a job is set to the maximum per-job VM charge in the system. Operational costs are set according to the electricity prices at 10 different geographic locations provided in [17], which vary on a hourly basis. Each server consumes power at 1 KW/h.

The number of job arrivals in each hour to the federation is set according to the cumulated job requests of each type submitted to the Google cluster during that hour, in the rough range of $[40000, 90000]$ requests per hour. We randomly assign each arrived job to one of the 10 clouds, following a heavy-tailed distribution. In operating the virtual queues, we set $\epsilon_s = 1000$ for jobs requiring low response delay, and $\epsilon_s = 500$ for those of long delays. The maximum number of job drops per hour is 1000 for all job types.

For comparison purposes, we also implement a simpler heuristic algorithm for each cloud to bid in the double auction and to schedules its jobs/servers: The cloud decides a value for each unscheduled job in a queue as the penalty to drop it if the next time slot is the deadline for scheduling, or the charged price upon its arrival otherwise. The true values of buy/sell prices for a type-m VM at this cloud are set to the same, as the largest average value of jobs in a queue, among all job queues requiring type-m VM(s). The quantity of VMs in a buy-bid is set to the number of unscheduled jobs in the queue with the largest average value as computed above. The quantity of VMs in a sell-bid is the overall number of VMs of the type that the cloud can provide. All VMs purchased via the auction are used to serve jobs from the queue with the maximum average value. A cloud maintains the minimum number of servers to support those jobs, and only drops a job when its maximum response delay is reached.



(a) $V = 4 \times 10^6$ (b) $V = 5 \times 10^6$

(c) $V = 6 \times 10^6$ (d) $V = 7 \times 10^6$

(e) $V = 8 \times 10^6$ (f) $V = 9 \times 10^6$

Fig. 2. Comparisons of individual profit with different values of V.

B. Individual Profit and Social Welfare

We compare the time-averaged profit achieved at each cloud with our dynamic algorithm in Alg. 1 and with the heuristic algorithm, after the system has been running for 2000 hours. Fig. 2 shows that our algorithm can achieve a higher profit than the heuristic, at each of the 10 clouds, when the value of V is no less than 4×10^6. The observation is that when V is larger, the individual profit with our algorithm is even better, since it is closer to the offline optimum.

We next compare the social welfare achieved with Alg. 1, the heuristic, and the dynamic benchmark Alg. 2. Fig. 3 shows that social welfare achieved with Alg. 1 is mostly within 7.7% of that by the benchmark algorithm, even under our heterogenous settings. It outperforms the heuristic by 19.2%.

The social welfare is larger at larger V's in cases of both Alg. 1 and the benchmark algorithm, verifying Theorems 6 and 8 in that they approach the respective offline optimum when V grows.

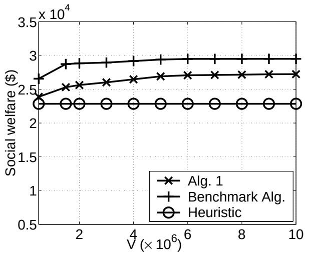
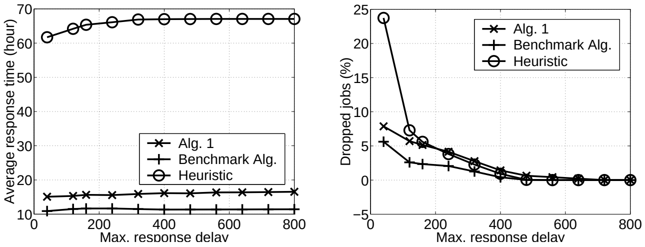


Fig. 3. Comparisons of social welfare.

C. Response Delay and Job Drop

We next investigate the scheduling delays experienced by jobs. In our system, a maximum response delay is set as the SLA objective for each type of jobs. Here, we study the average response delay actually experienced by the jobs, when the longer maximum response delay is set to different values. Fig. 4(a) shows that both Alg. 1 and the benchmark algorithm incur a low average response delay (well ahead of scheduling deadlines), as compared to that of the heuristic. The reasons are: i) the heuristic algorithm always greedily keeps jobs in queues for future scheduling until near the deadline; and ii) both Alg. 1 and the benchmark algorithm evaluate the scheduling urgency better than the heuristic does, such that jobs are tended to be served well before the deadlines.



(a) Average response delay (b) Averaged job-drop percentage

Fig. 4. Comparisons of average job scheduling delay and drop percentage.

We also study the percentage of admitted jobs in the entire federation that are eventually dropped with the three algorithms. Fig. 4(b) reveals that the drop rate decreases quickly with the increase of the allowed maximum response delay, and Alg. 1 and the benchmark algorithm again outperform the heuristic, due to their well-designed scheduling strategies.

IX. Conclusion

This paper investigates both individual-profit maximizing and social-welfare efficient strategies at individual selfish clouds in a cloud federation, in VM trades across cloud boundaries. We tailor a truthful, individual-rational, ex-post budget-balanced double auction as the inter-cloud trading mechanism, and design a dynamic algorithm for each cloud to decide the best VM valuation and bidding strategies, and to schedule job service/drop and server provisioning in the most economic fashion, under time-varying job arrivals and operational costs. The proposed algorithm can obtain a time-averaged profit for each cloud within a constant gap to its offline maximum, as well as a close-to-optimum social welfare in the entire federation, based on both solid theoretical analysis and trace-driven simulation studies under realistic setting. As future work, we are interested in broadening our investigations to front-end job pricing and competition for customers among the clouds, and the connection between front-end charging strategies and inter-cloud trading strategies in a cloud federation.

References

[1] B. Rochwerger, D. Breitgand, E. L. E, A. Galis, K. Nagin, I. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galn, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 54, pp. 535 – 545, 2009.

[2] E. Elmroth and L. Larsson, "Interfaces for placement , migration, and monitoring of virtual machines in federated clouds." in *Proc. of IEEE Computer Society GCC'09*, 2009.

[3] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Prof. of IEEE INFOCOM'10*, 2010.

[4] S. Ran, Y. He, and F. Xu, "Provably-efficient job scheduling for energy and fairness in geographically distributed data centers," in *Prof. of IEEE ICDCS'12*, 2012.

[5] R. Urgaonkar, U. Kozat, K. Igarashi, and M. Neely, "Dynamic resource allocation and power management in virtualized data centers," in *Prof. of IEEE/IFIP NOMS'10*, 2010.

[6] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. of IEEE INFOCOM'12*, 2012.

[7] R. Buyya, D. Abramson, and J. Giddy., "Nimrod/g: An architecture of a resource management and scheduling system in a global computational grid," in *Proc. of HPC Asia'00*, 2000.

[8] X. Zhou and H. Zheng, "Trust: A general framework for truthful double spectrum auctions," in *Proc. of IEEE INFOCOM'09*, 2009.

[9] H. Xu, J. Jin, and B. Li, "A secondary market for spectrum," in *Proc. of IEEE INFOCOM'10, Mini Conference*, 2010.

[10] [Online]. Available: http://aws.amazon.com/ec2

[11] M. Mihailescu and Y. M. Teo, "Dynamic resource pricing on federated clouds," in *Proc. of IEEE/ACM CCGrid'10*, 2010.

[12] ——, "The impact of user rationality in federated clouds," in *Proc. of IEEE/ACM CCGrid'12*, 2012.

[13] E. R. Gomes, Q. B. Vo, and R. Kowalczyk, "Pure exchange markets for resource sharing in federated clouds," *Concurrency Computat.: Pract. Exper.*, vol. 24, pp. 977 – 991, 2012.

[14] [Online]. Available: http://www.linode.com/faq.cfm

[15] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proc. of IEEE INFOCOM'11*, 2011.

[16] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines.* Morgan & Claypool, 2009.

[17] [Online]. Available: www.ferc.gov

[18] R. B. Myerson and M. A. Satterthwaite, "Efficient mechanisms for bilateral trading," *Journal of Economics Theory*, vol. 29, pp. 265–281, 1983.

[19] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, J. Walrand, Ed. Morgan&Claypool Publishers, 2010.

[20] T. SANDHOLM and S. SURI, "Market clearability," in *Proc. of IJICAI'01*, 2001.

[21] J. Wilkes, "More Google cluster data," Nov. 2011, URL: http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

[22] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema," Google Inc., Tech. Rep., 2011, revised 2012.03.20. URL: http://code.google.com/p/googleclusterdata/wiki/TraceVersion2.

[23] P. Huang, A. Scheller-Wolf, and K. Sycara, "Design of a multi-unit double auction e-market," *Computational Intelligence*, vol. 18, pp. 256–617, 2002.

Appendix A
Derivation of the one-shot optimization problem for individual profit maximization

Squaring the queuing laws (2) and (4), we can derive the following inequality

$$\begin{aligned}\Delta(\Theta_i(t)) \leq& \frac{1}{2}\sum_{s\in[1,S]}[[\sum_{j=1}^{F}\mu_{ij}^s(t)+D_i^s(t)]^2+[r_i^s(t)]^2+2Q_i^s(t)[r_i^s(t)\\ &-\sum_{j=1}^{F}\mu_{ij}^s(t)-D_i^s(t)]+[\mathbf{1}_{\{Q_i^s(t)>0\}}\epsilon_s]^2+[D_i^s(t)\\ &+\mathbf{1}_{\{Q_i^s(t)=0\}}\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s\\ &+\mathbf{1}_{\{Q_i^s(t)>0\}}\sum_{j=1}^{F}\mu_{ij}^s(t)]^2\\ &+2Z_i^s(t)[\mathbf{1}_{\{Q_i^s(t)>0\}}[\epsilon_s-\sum_{j=1}^{F}\mu_{ij}^s(t)]-D_i^s(t)\\ &-\mathbf{1}_{\{Q_i^s(t)=0\}}\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s]]\\ \leq& \frac{1}{2}\sum_{s\in[1,S]}[[\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s+D_i^{s(max)}]^2+[R_i^s]^2\\ &+2Q_i^s(t)[r_i^s(t)-\sum_{j=1}^{F}\mu_{ij}^s(t)-D_i^s(t)]\\ &+[\epsilon_s]^2+[D_i^{s(max)}+\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s]^2\\ &+2Z_i^s(t)[\epsilon_s-\sum_{j=1}^{F}\mu_{ij}^s(t)-D_i^s(t)]]\\ =& B_i+\sum_{s\in[1,S]}[Q_i^s(t)[r_i^s(t)-\sum_{j=1}^{F}\mu_{ij}^s(t)-D_i^s(t)]\\ &+Z_i^s(t)[\epsilon_s-\sum_{j=1}^{F}\mu_{ij}^s(t)-D_i^s(t)]],\end{aligned}$$

where $B_i=\frac{1}{2}\sum_{s\in[1,S]}[[\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s+D_i^{s(max)}]^2+[R_i^s]^2+[\epsilon_s]^2+[D_i^{s(max)}+\sum_{j=1}^{F}C_j^{m_s}N_j^{m_s}/g_s]^2]$.

By applying the drift-plus-penalty framework (or equivalently, drift-minus-profit here), we subtract the weighted one-shot individual profit of cloud i in time t, *i.e.*, $V\cdot[\sum_{m\in[1,M]}[\hat{s}_i^m(t)\hat{\eta}_i^m(t)-\hat{b}_i^m(t)\hat{\gamma}_i^m(t)-\beta_i(t)n_i^m(t)]+\sum_{s\in[1,S]}[p_i^s(t)\cdot r_i^s(t)-D_i^s(t)\xi_i^s]]$, on both sides of the above inequality. Hence, we have the following inequality:

$$\begin{aligned}&\Delta(\Theta_i(t))-V\cdot[\sum_{m\in[1,M]}[\hat{s}_i^m(t)\hat{\eta}_i^m(t)-\hat{b}_i^m(t)\hat{\gamma}_i^m(t)-\beta_i(t)n_i^m(t)]\\ &+\sum_{s\in[1,S]}[p_i^s(t)\cdot r_i^s(t)-D_i^s(t)\xi_i^s]]\\ \leq& B_i+\sum_{s\in[1,S]}[Q_i^s(t)r_i^s(t)+Z_i^s(t)\epsilon_s-Vp_i^s(t)\cdot r_i^s(t)]\\ &-\varphi_1^i(t)-\varphi_2^i(t)-\varphi_3^i(t),\end{aligned}$$

where $V>0$ is a user-defined positive constant that can be understood as the weight of profit in the expression.

Appendix B
Finding true values $\hat{b}_i^m(t)$, $\hat{\gamma}_i^m(t)$, $\hat{s}_i^m(t)$ and $\hat{\eta}_i^m(t)$

Based on individual rationality and truthfulness of the double auction mechanism, each buyer/seller pays/charges a price that is no higher/lower than the corresponding bid (true) value, while the number of VMs actually traded is no larger than the bid (true) value if the bid is successful, *i.e.*, $\hat{b}_i^m(t)\leq\tilde{b}_i^m(t)$, $\hat{s}_i^m(t)\geq\tilde{s}_i^m(t)$, $\hat{\gamma}_i^m(t)\leq\tilde{\gamma}_i^m(t)$ and $\hat{\eta}_i^m(t)\leq\tilde{\eta}_i^m(t)$. That is, the utility obtained by each cloud by participating in the auction is non-negative. Hence, the utility obtained by trading each VM at each winning buyer, *i.e.*, $\tilde{b}_i^m(t)-\hat{b}_i^m(t)$, or seller, *i.e.*, $\tilde{s}_i^m(t)-\hat{s}_i^m(t)$, is non-negative. Therefore, bidding for the maximum number of potential VMs provisioned, maximizes the utility of a seller or buyer, *i.e.*, the maximum number of type-m VMs a cloud is willing to sell or buy is the maximum number of potential type-m VMs provisioned in the federation, and hence the true values of the VM volumes to bid at each cloud are derived as in Eqn. (25) and (26), respectively.

We next identify the true values of the bidding prices for each type of VMs, $m\in[1,M]$, at cloud i case by case:

▷ *Case 1: Cloud i's buy-bid for type-m VMs wins, but not the sell-bid.*

In this case, we know that: i) all bought type-m VMs are from other clouds and should be used for job scheduling according to constraint (9); ii) $\hat{s}_i^m(t)=0$ and $\hat{\eta}_i^m(t)=0$.

A nice property of problem (19) is that, all decision variables related to type-m VMs, *i.e.*, $b_i^m(t)$, γ_i^m, $\hat{b}_i^m(t)$, $\hat{\gamma}_i^m(t)$, $\alpha_{ij}^m(t)$, $n_i^m(t)$, and $\mu_{ij}^s(t)$ with $m_s=m$, are independent from those related to the other types of VMs. Hence, the optimal solutions to decision variables related to type-m VMs can be derived by solving the following sub problem from (19):

$$\begin{aligned}\max\quad&\sum_{s:m_s=m,s\in[1,S]}\sum_{j\in[1,F]}\mu_{ij}^s(t)[Q_i^s(t)+Z_i^s(t)]-V\hat{b}_i^m(t)\hat{\gamma}_i^m(t)\\ &-V\beta_i(t)n_i^m(t) \qquad (47)\\ s.t.\quad&\text{Constraint (5)-(9).}\end{aligned}$$

In (47), we replace $\hat{\gamma}_i^m(t)$ by $\sum_{j\in[1,F]}\alpha_{ij}^m(t)$ based on Eqn. (7), and replace $\mu_{ij}^s(t)$'s by the optimal solutions in Eqn. (28) and (29) (to be derived in Sec. IV-C). We obtain

$$\begin{aligned}\max\quad&\sum_{j\neq i,j\in[1,F]}\alpha_{ij}^m(t)[\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}}-V\hat{b}_i^m(t)]\\ &+\mu_{ii}^{s_m^*}(t)(Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t))-V\beta_i(t)n_i^m(t). \qquad (48)\end{aligned}$$

According to Eqn. (28) and Eqn. (27), if $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}>\frac{\beta_i(t)}{C_i^m}$, we have

$$\begin{aligned}&\mu_{ii}^{s_m^*}(t)(Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t))-V\beta_i(t)n_i^m(t)\\ =&N_i^m[C_i^m\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}}-V\beta_i(t)];\end{aligned}$$

otherwise, we have

$$\mu_{ii}^{s_m^*}(t)(Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t))-V\beta_i(t)n_i^m(t)=0.$$

Both RHS values of the above equations are constants. As a result, the optimization problem (48) is finally equivalent to the following one:

$$\max \quad \sum_{j\neq i, j\in[1,F]} \alpha_{ij}^m(t)[\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}} - V\hat{b}_i^m(t)]. \quad (49)$$

According to the definition of true values, we know that the true value of $\tilde{b}_i^m(t)$ should be $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$ as defined in Eqn. (21), since: i) if $\hat{b}_i^m(t) > \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$, the utility in (49) is negative, and hence a profit loss in terms of problem (18) for cloud i; ii) if $\hat{b}_i^m(t) < \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$, the utility in (49) is positive, and hence a profit gain in terms of problem (18); and iii) if $\hat{b}_i^m(t) = \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$, the utility in (49) is zero, and the profit of cloud i in (18) remains the same as not acquiring the VMs.

▷ *Case 2: Cloud i's sell-bid for type-m VMs wins, but not the buy-bid.*

In this case, we know that: i) all type-m VMs sold from cloud i are used by other clouds for job scheduling according to constraint (9); and ii) $\hat{b}_i^m(t) = 0$ and $\hat{\gamma}_i^m(t) = 0$.

Similar to the analysis in Case 1, the optimal solutions to variables related to type-m VMs can be obtained by solving the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{s:m_s=m, s\in[1,S]} \sum_{j\neq i} \mu_{ij}^s(t)[Q_i^s(t)+Z_i^s(t)] + V\hat{s}_i^m(t)\hat{\eta}_i^m(t) \\ & - V\beta_i(t)n_i^m(t) \qquad (50) \\ s.t. \quad & \text{Constraint (5)-(9).} \end{aligned}$$

In (50), we replace $\hat{\eta}_i^m(t)$ by $\sum_{j\neq i}\alpha_{ji}^m(t)$ based on Eqn. (8) and the fact in this case that $\alpha_{ii}^m(t) = 0$, and replace $\mu_{ij}^s(t)$'s and $n_i^m(t)$'s with the optimal solutions in Eqn. (28), (29) and Eqn. (27). Then we have the following two cases:

i) if $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}} > \frac{\beta_i(t)}{C_i^m}$, problem (50) is equivalent to

$$\begin{aligned} \max \quad & N_i^m[C_i^m\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}} - V\beta_i(t)] \\ & + \sum_{j\neq i}\alpha_{ji}^m(t)[V\hat{s}_i^m(t) - \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}}], \end{aligned}$$

where the true value of $\tilde{s}_i^m(t)$ should be $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$ according to the definition of true value of the price to sell a type-m VM;

ii) otherwise, problem (50) is equivalent to

$$\max \quad V\sum_{j\neq i}\alpha_{ji}^m(t)[\hat{s}_i^m(t) - \beta_i(t)/C_i^m],$$

where the true value of $\tilde{s}_i^m(t)$ should be $\beta_i(t)/C_i^m$ according to the definition of the true value of the price to sell a type-m VM. Hence, we have derived the true values of $\tilde{s}_i^m(t)$ given in Eqn. (22).

▷ *Case 3: Both Cloud i's buy-bid and sell-bid for type-m VMs win.*

In this case, the following properties hold:

Property 1. If both cloud i's buy-bid and sell-bid for type-m VMs win, the cloud cannot buy a type-m VM with a price strictly higher than its price to sell a type-m VM, *i.e.*, $\hat{s}_i^m(t) \geq \hat{b}_i^m(t)$. Otherwise, there will be a positive profit loss at the cloud by self-trading its own type-m VMs, which violates its individual rationality.

Property 2. If both cloud i's buy-bid and sell-bid for type-m VMs win, the cloud cannot sell a type-m VM with a price strictly higher than its price to buy a type-m VM, *i.e.*, $\hat{s}_i^m(t) \leq \hat{b}_i^m(t)$. Otherwise, the auctioneer has to pay a positive sum to compensate for the price difference for those inter-cloud traded type-m VMs, which contradicts the ex-post budget balance property at the auctioneer.

Property 3. Based on Properties 1 and 2, if both cloud i's buy-bid and sell-bid for type-m VMs win, the actual buy and sell prices at the cloud for type-m VMs are the same, *i.e.*, $\hat{s}_i^m(t) = \hat{b}_i^m(t)$.

According to Property 3, we derive that the overall profit gain at cloud i for self-trading of type-m VMs, $\alpha_{ii}^m(t)[\hat{s}_i^m(t) - \hat{b}_i^m(t)]$, is zero.

The optimal solutions to variables related to type-m VMs can be obtained by solving the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{s:m_s=m, s\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^s(t)[Q_i^s(t)+Z_i^s(t)] - V\hat{b}_i^m(t)\hat{\gamma}_i^m(t) \\ & + V\hat{s}_i^m(t)\hat{\eta}_i^m(t) - V\beta_i(t)n_i^m(t) \qquad (51) \\ s.t. \quad & \text{Constraint (5)-(9).} \end{aligned}$$

In (51), we replace $\hat{\gamma}_i^m(t)$ by $\sum_{j\in[1,F]}\alpha_{ij}^m(t)$ based on Eqn. (7), and $\hat{\eta}_i^m(t)$ by $\sum_{j\in[1,F]}\alpha_{ji}^m(t)$ based on Eqn. (8). We also replace $\mu_{ij}^s(t)$'s and $n_i^m(t)$'s with the optimal solutions in Eqn. (28), (29) and Eqn. (27). Then we have the following two cases:

i) if $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}} > \frac{\beta_i(t)}{C_i^m}$, problem (51) is equivalent to

$$\begin{aligned} \max \quad & N_i^m[C_i^m\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}} - V\beta_i(t)] \\ & \sum_{j\neq i}\alpha_{ij}^m(t)[\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}} - V\hat{b}_i^m(t)] \\ & + \sum_{j\neq i}\alpha_{ji}^m(t)[V\hat{s}_i^m(t) - \frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}}], \end{aligned}$$

where the true values of $\tilde{b}_i^m(t)$ and $\tilde{s}_i^m(t)$ should both be $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$ according to the definition of true value of the price to buy/sell a type-m VM;

ii) otherwise, problem (51) is equivalent to

$$\begin{aligned} \max \quad & \sum_{j\neq i}\alpha_{ij}^m(t)[\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{g_{s_m^*}} - V\hat{b}_i^m(t)] \\ & V\sum_{j\neq i}\alpha_{ji}^m(t)[\hat{s}_i^m(t) - \beta_i(t)/C_i^m], \end{aligned}$$

where the true value of $\tilde{b}_i^m(t)$ should be $\frac{Q_i^{s_m^*}(t)+Z_i^{s_m^*}(t)}{Vg_{s_m^*}}$, and the true value of $\tilde{s}_i^m(t)$ should be $\beta_i(t)/C_i^m$ according to the definition of the true value of the price to buy/sell a type-m VM. Hence, we have derived the true values of $\tilde{b}_i^m(t)$ and $\tilde{s}_i^m(t)$ given in Eqn. (21) and Eqn. (22).

Appendix C
Derivation of the one-shot optimization problem for social welfare maximization

Squaring the queuing laws (2) and (4), we can derive the following inequality

$$
\begin{aligned}
\Delta(\Theta(t)) \leq & \frac{1}{2} \sum_{i\in[1,F]} \sum_{s\in[1,S]} [[\sum_{j=1}^{F} \mu_{ij}^{s}(t) + D_i^s(t)]^2 + [r_i^s(t)]^2 \\
& + 2Q_i^s(t)[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s}(t) - D_i^s(t)] + [\mathbf{1}_{\{Q_i^s(t)>0\}}\epsilon_s]^2 \\
& + [D_i^s(t) + \mathbf{1}_{\{Q_i^s(t)=0\}} \sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s \\
& + \mathbf{1}_{\{Q_i^s(t)>0\}} \sum_{j=1}^{F} \mu_{ij}^{s}(t)]^2 \\
& + 2Z_i^s(t)[\mathbf{1}_{\{Q_i^s(t)>0\}}[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s}(t)] - D_i^s(t) \\
& - \mathbf{1}_{\{Q_i^s(t)=0\}} \sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s]] \\
\leq & \frac{1}{2} \sum_{i\in[1,F]} \sum_{s\in[1,S]} [[\sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s + D_i^{s(max)}]^2 + [R_i^s]^2 \\
& + 2Q_i^s(t)[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s}(t) - D_i^s(t)] \\
& + [\epsilon_s]^2 + [D_i^{s(max)} + \sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s]^2 \\
& + 2Z_i^s(t)[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s}(t) - D_i^s(t)]] \\
= & B + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s}(t) - D_i^s(t)] \\
& + Z_i^s(t)[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s}(t) - D_i^s(t)]],
\end{aligned}
$$

where $B = \frac{1}{2}\sum_{i\in[1,F]}\sum_{s\in[1,S]}[[\sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s + D_i^{s(max)}]^2 + [R_i^s]^2 + [\epsilon_s]^2 + [D_i^{s(max)} + \sum_{j=1}^{F} C_j^{m_s} N_j^{m_s}/g_s]^2]$.

By applying the drift-plus-penalty framework (or equivalently, drift-minus-profit here), we subtract the weighted one-shot social welfare in time t, *i.e.*, $V \cdot \sum_{i\in[1,F]}[\sum_{m\in[1,M]}[\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)] + \sum_{s\in[1,S]}[p_i^s(t) \cdot r_i^s(t) - D_i^s(t)\xi_i^s]]$, on both sides of the above inequality. Hence, we have the following inequality:

$$
\begin{aligned}
& \Delta(\Theta(t)) - V \cdot \sum_{i\in[1,F]} [\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) \\
& - \beta_i(t)n_i^m(t)] + \sum_{s\in[1,S]} [p_i^s(t) \cdot r_i^s(t) - D_i^s(t)\xi_i^s]] \\
\leq & B + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)r_i^s(t) + Z_i^s(t)\epsilon_s - Vp_i^s(t) \cdot r_i^s(t)] \\
& - \varphi_1(t) - \varphi_2(t),
\end{aligned}
$$

where $V > 0$ is a user-defined positive constant that can be understood as the weight of profit in the expression.

Appendix D
Proof to Theorem 2

This theorem can be proved based on the following two lemmas.

Lemma 2 (Monotonic winner determination): Given prices of buy-bids $\{b_1^m(t), \ldots, b_i^m(t), \ldots, b_F^m(t)\}$ and sell-bids $\{s_1^m(t), \ldots, s_i^m(t), \ldots, s_F^m(t)\}$, we have that

1) If cloud i wins the buy-bid by bidding with $b_i^m(t)$, then cloud i also wins the buy-bid by bidding with $b' > b_i^m(t)$;
2) If cloud i wins the sell-bid by bidding with $s_i^m(t)$, then cloud i also wins the sell-bid by bidding with $s' < s_i^m(t)$.

Proof: We prove the cases in the lemma respectively as follows,

1) Since cloud i wins the buy-bid with $b_i^m(t)$, we know that $b_i^m(t)$ is the largest among all buy-bids, *i.e.*, $b_i^m(t) \geq b_j^m(t)$, $\forall j \neq i, j \in [1, F]$. With $b' > b_i^m(t)$, we have that $b' > b_j^m(t)$, $\forall j \neq i, j \in [1, F]$. Hence, if cloud i proposes a buy-bid with b', it can still win the buy-bid according to our winner determination decision, since its buy-bid price is still the largest among all buy-bids.
2) Since cloud i wins the sell-bid with $s_i^m(t)$ and the sell-bids are sorted in ascending order, we know that $s_i^m(t) \leq \vartheta_{j'}^m(t)$, where j' is the critical index as defined in Eqn. (31), and $s_i^m(t)$ is among the $(j'-1)^{th}$ lowest sell-bids in ascending order. With $s' < s_i^m(t)$, we also have that $s' < \vartheta_{j'}^m(t)$ and s' is among the $(j'-1)^{th}$ lowest sell-bids in ascending order. Hence, if cloud i propose a sell-bid with s', it can still win the sell-bid according to our winner determination decision, since its sell-bid price is still among the $(j'-1)^{th}$ lowest sell-bids.

■

Lemma 3 (Bid-independent pricing): Given prices of buy-bids $\{b_1^m(t), \ldots, b_i^m(t), \ldots, b_F^m(t)\}$ and sell-bids $\{s_1^m(t), \ldots, s_i^m(t), \ldots, s_F^m(t)\}$, we have that

1) If cloud i wins the buy-bid by bidding with $b_i^m(t)$ and b', the charged price $\hat{b}_i^m(t)$ to cloud i is the same for both;
2) If cloud i wins the sell-bid by bidding with $s_i^m(t)$ and s', the charged price $\hat{s}_i^m(t)$ to cloud i is the same for both.

Proof: We prove the the cases in the lemma respectively as follows,

1) Since cloud i wins the buy-bid, the charged price $\hat{b}_i^m(t)$ should be $\theta_2^m(t)$, which is the second largest buy-bid price independent of cloud i's buy-bid, according to our pricing scheme in the auction. And we have that $b_i^m(t) \geq \theta_2^m(t)$ and $b' \geq \theta_2^m(t)$. We also have that, as long as cloud i wins the buy-bid, the value of the second largest buy-bid price $\theta_2^m(t)$ does not change. Hence, cloud i should be charged with $\theta_2^m(t)$ by bidding with no matter $b_i^m(t)$ or b'.

2) Since cloud i wins the sell-bid, the charged price $\hat{s}_i^m(t)$ should be $\vartheta_{j'}^m(t)$, which is the j'^{th} lowest sell-bid price, according to our pricing scheme in the auction. And we have that $s_i^m(t) \leq \vartheta_{j'}^m(t)$ and $s' \leq \vartheta_{j'}^m(t)$. As long as cloud i wins the sell-bid, we have that $s_i^m(t)$ and s' should be among the $(j'-1)^{th}$ lowest sell-bid prices and the value of $\vartheta_{j'}^m(t)$ does not change. Hence, cloud i should be charged with the same price with $\vartheta_{j'}^m(t)$ by bidding with no matter $b_i^m(t)$ or b' if it wins the sell-bid.

■

We can then prove the Theorem 2 to show that any cloud i cannot obtain higher utility gain by bidding untruthfully, *i.e.*, $b_i^m(t) \neq \tilde{b}_i^m(t)$ and/or $s_i^m(t) \neq \tilde{s}_i^m(t)$, $\forall m \in [1, M]$, by analyzing all possible auction results.

Case 1 – *Cloud i wins both buy-bid and sell-bid with truthful bidding*: In this case, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively. We discuss the all possibly cases of untruthful bidding as follows,

- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the buy-bid but may either win or lose the sell-bid. If it also wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to Lemma 3 and our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)]\hat{\eta}_i^m(t)$ with $\vartheta_{j'}^m(t) \geq \tilde{s}_i^m(t)$, by losing the sell-bid.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) = \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the buy-bid. It is also easy to see that cloud i still wins the sell-bid with the same bidding price for sell-bid. Hence, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins both the buy-bid and the sell-bid. The charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: Cloud i stills wins the buy-bid with the same bidding price for buy-bid. However, it may either win or lose the sell-bid. If it also wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, by losing the sell-bid.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the sell-bid. It is also easy to see that cloud i still wins the buy-bid with the same bidding price for buy-bid. Hence, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: The cloud can either win both buy and sell bids, or win buy-bid only, or win sell-bid only, or lose both bids. If the cloud still wins both bids, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it only wins the buy-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to Lemma our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, by losing the sell-bid. If it only wins the sell-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme, and no VM is bought by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\tilde{b}_i^m(t) - \theta_2^m(t)] \cdot \hat{\gamma}_i^m(t)$ with $\tilde{b}_i^m(t) \geq \theta_2^m(t)$, by losing the buy-bid. If it loses both bids, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought by or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since there is non-negative utility loss, *i.e.*, $[\tilde{b}_i^m(t) - \theta_2^m(t)] \cdot \hat{\gamma}_i^m(t) + [\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)] \cdot \hat{\eta}_i^m(t)$ with $\tilde{b}_i^m(t) \geq \theta_2^m(t)$ and $\vartheta_{j'}^m(t) \geq \tilde{s}_i^m(t)$, by losing the both bids.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) = \tilde{s}_i^m(t)$*: Cloud i stills wins the sell-bid with the same bidding price for sell-bid. However, it may either win or lose the buy-bid. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme, and no VM is bought by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss,

by losing the buy-bid.

- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the sell-bid but may either win or lose the buy-bid. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme, and no VM is bought by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, by losing the buy-bid.

Case 2 – *Cloud i wins buy-bid but loses sell-bid with truthful bidding*: In this case, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively. we discuss the all possible cases of untruthful bidding as follows,

- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the buy-bid. It is also easy to see that cloud i still loses the sell-bid, since otherwise we will have a contradiction to Lemma 2. Hence, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to Lemma 3 and our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) = \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the buy-bid. It is also easy to see that cloud i still loses the sell-bid with the same bidding price for sell-bid. Hence, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to Lemma 3 and pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the buy-bid but may either win or lose the sell-bid. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to Lemma 3 and our pricing scheme, and no VM is sold by cloud i; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it also wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to Lemma 3 and our pricing scheme; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\tilde{s}_i^m(t) - \vartheta_{j'}^m(t)]\hat{\eta}_i^m(t)$ with $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, by winning the sell-bid.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: Cloud i still wins the buy-bid with the same bidding price. It is also easy to see that cloud i still loses the sell-bid, since otherwise we will have a contradiction to Lemma 2. Hence, the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: Cloud i still wins the buy-bid with the same bidding price. However, it may either win or lose the sell-bid. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it also wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, by winning the sell-bid.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: It is easy to see that cloud i still loses the sell-bid, since otherwise we will have a contradiction to Lemma 2. However, it may either win or lose the buy-bid. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are 0 and 0, respectively, according to Lemma our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\tilde{b}_i^m(t) - \theta_2^m(t)] \cdot \hat{\gamma}_i^m(t)$ with $\tilde{b}_i^m(t) \geq \theta_2^m(t)$, by losing the buy-bid.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) = \tilde{s}_i^m(t)$*: The cloud i still loses the sell-bid with the same bidding price. However, it may either win or lose the buy-bid. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are 0 and 0, respectively, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, by losing the buy-bid.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: The cloud can either win both buy and sell bids, or win buy-bid only, or win sell-bid only, or lose both bids. If the cloud wins both bids, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged buy-bid price remains the same while

there is non-negative utility loss, by winning the sell-bid. If it only wins the buy-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it only wins the sell-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to our pricing scheme, and no VM is bought by cloud i; the utility gain is non-positive by bidding untruthfully, since there is non-negative utility loss, *i.e.*, $[\tilde{b}_i^m(t) - \theta_2^m(t)] \cdot \hat{\gamma}_i^m(t) - [\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)] \cdot \hat{\eta}_i^m(t)$ with $\tilde{b}_i^m(t) \geq \theta_2^m(t)$ and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, by losing the buy-bid while winning the sell-bid. If it loses both bids, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought by or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\tilde{b}_i^m(t) - \theta_2^m(t)] \cdot \hat{\gamma}_i^m(t)$ with $\tilde{b}_i^m(t) \geq \theta_2^m(t)$, by losing the buy-bid.

Case 3 – *Cloud i wins sell-bid but loses buy-bid with truthful bidding*: In this case, the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively. we discuss the all possible cases of untruthful bidding as follows,

- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: The cloud can either win both buy and sell bids, or win buy-bid only, or win sell-bid only, or lose both bids. If the cloud wins both bids, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t) \geq \tilde{b}_i^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully since the charged sell-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\theta_2^m(t) - \tilde{b}_i^m(t)] \cdot \hat{\gamma}_i^m(t)$ with $\tilde{b}_i^m(t) \leq \theta_2^m(t)$, by winning the buy-bid. If it only wins the buy-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully since there is non-negative utility loss, *i.e.*, $[\theta_2^m(t) - \tilde{b}_i^m(t)] \cdot \hat{\gamma}_i^m(t) + [\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)] \cdot \hat{\eta}_i^m(t)$ with $\tilde{b}_i^m(t) \leq \theta_2^m(t)$ and $\vartheta_{j'}^m(t) \geq \tilde{s}_i^m(t)$, by winning the buy-bid while losing the sell-bid. If it only wins the sell-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to our pricing scheme, and no VM is bought by cloud i; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses both bids, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought by or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)] \cdot \hat{\eta}_i^m(t)$ with $\vartheta_{j'}^m(t) \geq \tilde{s}_i^m(t)$, by losing the sell-bid.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) = \tilde{s}_i^m(t)$*: Cloud i still wins the sell-bid with the same bidding price. However, it may either win or lose the buy-bid. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully since the charged sell-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\theta_2^m(t) - \tilde{b}_i^m(t)] \cdot \hat{\gamma}_i^m(t)$ with $\tilde{b}_i^m(t) \leq \theta_2^m(t)$, by winning the buy-bid. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged/paid prices are the same with that by bidding truthfully.
- *Bid untruthfully with $b_i^m(t) > \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the sell-bid but may either win or lose the buy-bid. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are $\theta_2^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme, and no VM is sold by cloud i; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta_2^m(t) \geq \tilde{b}_i^m(t)$ and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme; the utility gain is non-positive by bidding untruthfully, since the charged sell-bid price remains the same while there is non-negative utility loss, by winning the buy-bid.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: Cloud i still loses the buy-bid with the same bidding price. However, it may either win or lose the sell-bid. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are 0 and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully since the charged buy-bid price remains the same while there is non-negative utility loss, *i.e.*, $[\vartheta_{j'}^m(t) - \tilde{s}_i^m(t)] \cdot \hat{\eta}_i^m(t)$ with $\vartheta_{j'}^m(t) \geq \tilde{s}_i^m(t)$, by losing the sell-bid. If it also wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t) \leq \tilde{s}_i^m(t)$, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully.
- *Bid untruthfully with $b_i^m(t) = \tilde{b}_i^m(t)$ and $s_i^m(t) < \tilde{s}_i^m(t)$*: According to Lemma 2, cloud i still wins the sell-bid. It is also easy to see that cloud i still loses the buy-bid with the same bidding price. Hence, the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t)$, respectively, according to Lemma 3 and our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b_i^m(t) < \tilde{b}_i^m(t)$ and $s_i^m(t) > \tilde{s}_i^m(t)$*: It is easy to see that cloud i still loses the buy-bid,

since otherwise we will have a contradiction to Lemma 2. However, it may either win or lose the sell-bid. If it wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta^m_{j'}(t)$, respectively, according to our pricing scheme; the utility gain is zero by bidding untruthfully since the charged/paid prices are the same with that by bidding truthfully. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is non-positive by bidding untruthfully, since the charged buy-bid price remains the same while there is non-negative utility loss, by losing the sell-bid.

- *Bid untruthfully with $b^m_i(t) < \tilde{b}^m_i(t)$ and $s^m_i(t) = \tilde{s}^m_i(t)$*: Cloud i still wins the sell-bid with the same bidding price. It is also easy to see that cloud i still loses the buy-bid, since otherwise we will have a contradiction to Lemma 2. Hence, the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta^m_{j'}(t)$, respectively, according to our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b^m_i(t) < \tilde{b}^m_i(t)$ and $s^m_i(t) < \tilde{s}^m_i(t)$*: According to Lemma 2, cloud i still wins the sell-bid. It is also easy to see that cloud i still loses the buy-bid, since otherwise we will have a contradiction to Lemma 2. Hence, the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta^m_{j'}(t)$, respectively, according to Lemma 3 and our pricing scheme. The utility gain is zero by bidding untruthfully.

Case 4 – *Cloud i loses both buy-bid and sell-bid with truthful bidding*: In this case, the charged/paid prices for buy-bid and sell-bid are both 0. we discuss the all possible cases of untruthful bidding as follows,

- *Bid untruthfully with $b^m_i(t) > \tilde{b}^m_i(t)$ and $s^m_i(t) > \tilde{s}^m_i(t)$*: It is easy to see that cloud i still loses the sell-bid, since otherwise we will have a contradiction to Lemma 2. However, it may either win or lose the buy-bid. If it wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta^m_2(t) \geq \tilde{b}^m_i(t)$ and 0, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully there is non-negative utility loss, *i.e.*, $[\theta^m_2(t) - \tilde{b}^m_i(t)] \cdot \hat{\gamma}^m_i(t)$ with $\tilde{b}^m_i(t) \leq \theta^m_2(t)$, by winning the buy-bid. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b^m_i(t) > \tilde{b}^m_i(t)$ and $s^m_i(t) = \tilde{s}^m_i(t)$*: Cloud i still loses the sell-bid with the same bidding price. However, it may either win or lose the buy-bid. If it loses the buy-bid, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is zero by bidding untruthfully. If it also wins the buy-bid, we have that the charged/paid prices for buy-bid and sell-bid are $\theta^m_2(t) \geq \tilde{b}^m_i(t)$ and 0, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully there is non-negative utility loss, by winning the buy-bid.
- *Bid untruthfully with $b^m_i(t) > \tilde{b}^m_i(t)$ and $s^m_i(t) < \tilde{s}^m_i(t)$*: The cloud can either win both buy and sell bids, or win buy-bid only, or win sell-bid only, or lose both bids. If the cloud wins both bids, we have that the charged/paid prices for buy-bid and sell-bid are $\theta^m_2(t) \geq \tilde{b}^m_i(t)$ and $\vartheta^m_{j'}(t) \leq \tilde{s}^m_i(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully since there is non-negative utility loss, *i.e.*, $[\theta^m_2(t) - \tilde{b}^m_i(t)] \cdot \hat{\gamma}^m_i(t) + [\tilde{s}^m_i(t) - \vartheta^m_{j'}(t)] \cdot \hat{\eta}^m_i(t)$ with $\tilde{b}^m_i(t) \leq \theta^m_2(t)$ and $\vartheta^m_{j'}(t) \leq \tilde{s}^m_i(t)$, by winning the buy-bid and sell-bid. If it only wins the buy-bid, we have that the charged prices for buy-bid and sell-bid are $\theta^m_2(t)$ and 0, respectively, according to our pricing scheme, and no VM is sold by cloud i; the utility gain is non-positive by bidding untruthfully since the paid price for sell-bid is the same with that by bidding truthfully and there is non-negative utility loss, by winning the buy-bid. If it only wins the sell-bid, we have that the charged prices for buy-bid and sell-bid are 0 and $\vartheta^m_{j'}(t) \leq \tilde{s}^m_i(t)$, respectively, according to our pricing scheme, and no VM is bought by cloud i; the utility gain is non-positive by bidding untruthfully since the charged price for the buy-bid is the same with that by bidding truthfully while there is non-negative utility loss, *i.e.*, $[\tilde{s}^m_i(t) - \vartheta^m_{j'}(t)] \cdot \hat{\eta}^m_i(t)$ with $\vartheta^m_{j'}(t) \leq \tilde{s}^m_i(t)$, by winning the sell-bid. If it loses both bids, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought by or sold by cloud i; the utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b^m_i(t) = \tilde{b}^m_i(t)$ and $s^m_i(t) > \tilde{s}^m_i(t)$*: It is easy to see that cloud i still loses the sell-bid, since otherwise we will have a contradiction to Lemma 2. It is also not hard to find that cloud i still loses the buy-bid with the same bidding price. Hence, the charged/paid prices for buy-bid and sell-bid are both 0, according to our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b^m_i(t) = \tilde{b}^m_i(t)$ and $s^m_i(t) < \tilde{s}^m_i(t)$*: It is not hard to find that cloud i still loses the buy-bid with the same bidding price. However, it may either win or lose the sell-bid. If it wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta^m_{j'}(t) \leq \tilde{s}^m_i(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully since the charged price for buy-bid is the same with that by bidding truthfully while there is non-negative utility loss, by winning the sell-bid. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is zero by bidding untruthfully.
- *Bid untruthfully with $b^m_i(t) < \tilde{b}^m_i(t)$ and $s^m_i(t) > \tilde{s}^m_i(t)$*:

It is easy to see that cloud i still loses both the buy-bid and the sell-bid, since otherwise we will have a contradiction to Lemma 2. Hence, the charged/paid prices for buy-bid and sell-bid are both 0, according to our pricing scheme. The utility gain is zero by bidding untruthfully.

- *Bid untruthfully with* $b_i^m(t) < \tilde{b}_i^m(t)$ *and* $s_i^m(t) = \tilde{s}_i^m(t)$: It is easy to see that cloud i still loses the buy-bid, since otherwise we will have a contradiction to Lemma 2. It is also not hard to find that cloud i still loses the sell-bid with the same bidding price. Hence, the charged/paid prices for buy-bid and sell-bid are both 0, according to our pricing scheme. The utility gain is zero by bidding untruthfully.
- *Bid untruthfully with* $b_i^m(t) < \tilde{b}_i^m(t)$ *and* $s_i^m(t) < \tilde{s}_i^m(t)$: It is easy to see that cloud i still loses the buy-bid, since otherwise we will have a contradiction to Lemma 2. However, it may either win or lose the sell-bid. If it wins the sell-bid, we have that the charged/paid prices for buy-bid and sell-bid are 0 and $\vartheta_{j'}^m(t) \le \tilde{s}_i^m(t)$, respectively, according to our pricing scheme; the utility gain is non-positive by bidding untruthfully since the charged price for buy-bid is the same with that by bidding truthfully while there is non-negative utility loss, by winning the sell-bid. If it loses the sell-bid, we have that the charged prices for buy-bid and sell-bid are both 0, according to our pricing scheme, and no VM is bought or sold by cloud i; the utility gain is zero by bidding untruthfully.

To conclude, we have shown that bidding truthfully is the dominant strategy of each cloud.

Appendix E
Proof to Theorem 3

We prove the individual rationality for buy-bids and sell-bids, respectively.

Winner of buy-bid: If cloud i wins the buy-bid for VM type m, we have that $b_i^m(t)$ is the largest buy-bid price among all buy-bids, and $b_i^m(t) \ge \theta_2^m(t)$, according to our winner determination scheme. Since $\hat{b}_i^m(t) = \theta_2^m(t)$ according to our pricing scheme, we have that $\hat{b}_i^m(t) \le b_i^m(t)$.

Winner of sell-bid: If cloud i wins the sell-bid for VM type m, we have that $s_i^m(t)$ is among the $(j'-1)^{th}$ lowest sell-bid prices of all sell-bids, and $s_i^m(t) \le \vartheta_{j'}^m(t)$ according to our winner determination scheme. Since $\hat{s}_i^m(t) = \vartheta_{j'}^m(t)$ according to our pricing scheme, we have that $\hat{s}_i^m(t) \ge s_i^m(t)$.

Appendix F
Proof to Theorem 4

We first calculate the total payment from the buyers and the total price paid to the sellers, respectively. We then show that the ex-post budget balance is guaranteed.

Total payment from the buyers: According to our winner determination scheme, only the buyer with largest bidding price wins the buy-bid. With the pricing scheme in Eqn. (32), the charged price is $\theta_2^m(t)$ for each bought VM. With the allocation scheme in Eqn. (34), the overall number of bought VMs is $\sum_{j=1}^{j'-1} L_j^m(t)$. Hence, the total payment from the buyers is that

$$\sum_{i\in[1,F]} [\hat{b}_i^m(t) \cdot \hat{\gamma}_i^m(t)] = \theta_2^m(t) \cdot \sum_{j=1}^{j'-1} L_j^m(t).$$

Overall price paid to the sellers: According to our winner determination scheme, only the seller with $(j'-1)^{th}$ lowest bidding price wins the sell-bid. With the pricing scheme in Eqn. (33), the paid price is $\vartheta_{j'}^m(t)$ for each sold VM. With the allocation scheme in Eqn. (35), the overall number of sold VMs is $\sum_{j=1}^{j'-1} L_j^m(t)$. Hence, the total payment to the sellers is that

$$\sum_{i\in[1,F]} [\hat{s}_i^m(t) \cdot \hat{\eta}_i^m(t)] = \vartheta_{j'}^m(t) \cdot \sum_{j=1}^{j'-1} L_j^m(t).$$

According to Eqn. (31) for the winner determination scheme, we know that $\vartheta_{j'}^m(t) \le \theta_2^m(t)$. Hence, the ex-post budget balance at the auctioneer for each VM type $m \in [1, M]$ can be guaranteed as follows,

$$\begin{aligned}
&\sum_{i\in[1,F]} [\hat{b}_i^m(t) \cdot \hat{\gamma}_i^m(t) - \hat{s}_i^m(t) \cdot \hat{\eta}_i^m(t)] \\
=&\sum_{i\in[1,F]} [\hat{b}_i^m(t) \cdot \hat{\gamma}_i^m(t)] - \sum_{i\in[1,F]} [\hat{s}_i^m(t) \cdot \hat{\eta}_i^m(t)] \\
=&\theta_2^m(t) \cdot \sum_{j=1}^{j'-1} L_j^m(t) - \vartheta_{j'}^m(t) \cdot \sum_{j=1}^{j'-1} L_j^m(t) \\
\ge&0.
\end{aligned}$$

Appendix G
Proof to Lemma 1

We prove the lemma by induction.

Induction Basis: At time slot 0, the beginning of the federation, all queues are empty. Therefore,

$$Q_i^s(0) = 0 \le Q_i^{s(max)}, \ \forall i \in [1,F], s \in [1,S],$$
$$Z_i^s(0) = 0 \le Z_i^{s(max)}, \ \forall i \in [1,F], s \in [1,S].$$

Induction Step: Suppose that, at time slot $t \ge 0$, $Q_i^s(t) \le Q_i^{s(max)}$ and $Z_i^s(t) \le Z_i^{s(max)}$, $\forall i \in [1,F], s \in [1,S]$. Then, for any $Q_i^s(t)$ and $Z_i^s(t)$, we have the following possible cases.

- $0 \le Q_i^s(t) \le V\xi_i^s$ or $V\xi_i^s < Q_i^s(t) \le V\xi_i^s + R_i^s$;
- $0 \le Z_i^s(t) \le V\xi_i^s$ or $V\xi_i^s < Z_i^s(t) \le V\xi_i^s + \epsilon_s$.

- We first analyze the size of $Q_i^s(t+1)$:

- If $0 \le Q_i^s(t) \le V\xi_i^s$, we have that

$$\begin{aligned}
Q_i^s(t+1) &= \max\{Q_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^s(t) - D_i^s(t), 0\} + r_i^s(t) \\
&\le \max\{Q_i^s(t), 0\} + R_i^s \\
&\le V\xi_i^s + R_i^s = Q_i^{s(max)}
\end{aligned}$$

according to the queueing law (2). The first inequality is based on the fact that $0 \le r_i^s(t) \le R_i^s$.

- If $V\xi_i^s < Q_i^s(t) \le V\xi_i^s + R_i^s$, we have that

$$D_i^s(t) = D_i^{s(max)},$$

according to the job drop decision with Eqn. (30). Hence, we have that

$$\begin{aligned} Q_i^s(t+1) &= \max\{Q_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^s(t) - D_i^s(t), 0\} + r_i^s(t) \\ &\le \max\{V\xi_i^s + R_i^s - D_i^{s(max)}, 0\} + R_i^s \\ &\le V\xi_i^s + R_i^s \le Q_i^{s(max)}. \end{aligned}$$

The second inequality is based on the fact that $D_i^{s(max)} \ge \max\{R_i^s, \epsilon_s\}$.

So far, $Q_i^s(t) \le Q_i^{s(max)}$, $\forall i \in [1, F], s \in [1, S]$ for each time slot t is proved.

- We next analyze the size of $Z_i^s(t+1)$:

- If $0 \le Z_i^s(t) \le V\xi_i^s$, we have that

$$\begin{aligned} Z_i^s(t+1) = \max\{&Z_i^s(t) + \mathbf{1}_{\{Q_i^s(t)>0\}} \cdot [\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^s(t)] - D_i^s(t) \\ &- \mathbf{1}_{\{Q_i^s(t)=0\}} \cdot \sum_{j=1}^{F} \frac{C_j^{m_s} \cdot N_j^{m_s}}{g_s}, 0\} \\ \le& \max\{Z_i^s(t) + \epsilon_s, 0\} \\ \le& V\xi_i^s + \epsilon_s = Z_i^{s(max)}, \end{aligned}$$

  according to the queueing law (4). The first inequality is based on the fact that $\epsilon_s > 0$.

- If $V\xi_i^s < Z_i^s(t) \le V\xi_i^s + \epsilon_s$, we have that

$$D_i^s(t) = D_i^{s(max)},$$

  according to the job drop decision with Eqn. (30). Hence, we have that

$$\begin{aligned} Z_i^s(t+1) = \max\{&Z_i^s(t) + \mathbf{1}_{\{Q_i^s(t)>0\}} \cdot [\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^s(t)] - D_i^s(t) \\ &- \mathbf{1}_{\{Q_i^s(t)=0\}} \cdot \sum_{j=1}^{F} \frac{C_j^{m_s} \cdot N_j^{m_s}}{g_s}, 0\} \\ \le& \max\{Z_i^s(t) + \epsilon_s - D_i^{s(max)}, 0\} \\ \le& V\xi_i^s + \epsilon_s = Z_i^{s(max)}. \end{aligned}$$

  The second inequality is based on the fact that $D_i^{s(max)} \ge \max\{R_i^s, \epsilon_s\}$.

So far, $Z_i^s(t) \le Z_i^{s(max)}$, $\forall i \in [1, F], s \in [1, S]$ for each time slot t is proved.

In conclusion, Lemma 1 is proven.

Appendix H
Proof to Theorem 5

We prove this theorem by contradiction.

For each cloud $i \in [1, F]$ and each service type $s \in [1, S]$, the job requests arrive at time slot $t \ge 0$ is $r_i^s(t)$ and the earliest time they can depart the queue is $t+1$. We show that all these jobs depart (by being either scheduled or dropped) on or before time $t + d_s$.

Suppose this is not true, we will come to a contradiction. We must have that $Q_i^s(\tau) > 0$ for all $\tau \in [t+1, \ldots, t+d_s]$ (otherwise, all the jobs are scheduled by time $t+d_s$). With the queueing law in Eqn. (4), we have that

$$\begin{aligned} Z_i^s(\tau+1) &= \max\{Z_i^s(\tau) + \epsilon_s - \sum_{j=1}^{F} \mu_{ij}^s(\tau) - D_i^s(\tau), 0\} \\ &\ge Z_i^s(\tau) + \epsilon_s - \sum_{j=1}^{F} \mu_{ij}^s(\tau) - D_i^s(\tau). \end{aligned}$$

Summing the above over $\tau \in [t+1, \ldots, t+d_s]$, we have that

$$Z_i^s(t+d_s+1) - Z_i^s(t+1) \ge \epsilon_s d_s - \sum_{\tau=t+1}^{t+d_s} [\sum_{j=1}^{F} \mu_{ij}^s(\tau) + D_i^s(\tau)].$$

Rearranging the above inequality and using the fact that $Z_i^s(t+d_s+1) \le Z_i^{s(max)}$ and $Z_i^s(t+1) \ge 0$, we have that

$$\epsilon_s d_s - Z_i^{s(max)} \le \sum_{\tau=t+1}^{t+d_s} [\sum_{j=1}^{F} \mu_{ij}^s(\tau) + D_i^s(\tau)]. \tag{52}$$

Since the jobs are scheduled in a FIFO fashion, the jobs $r_i^s(t)$ that arrive at slot t are placed at the end of the queue at slot $t+1$, and should be fully cleared only when all the jobs backlogged in $Q_i^s(t+1)$ have been scheduled. That is, the last job of $r_i^s(t)$ departs on slot $t+T$ with $T > 0$ as the smallest integer satisfying $\sum_{\tau=t+1}^{t+T} [\sum_{j=1}^{F} \mu_{ij}^s(\tau) + D_i^s(\tau)] \ge Q_i^s(t+1)$. Based on our assumption that not all of the $r_i^s(t)$ jobs depart by time $t+d_s$, we must have that

$$\sum_{\tau=t+1}^{t+d_s} [\sum_{j=1}^{F} \mu_{ij}^s(\tau) + D_i^s(\tau)] < Q_i^s(t+1) \le Q_i^{s(max)}. \tag{53}$$

Combining Eqn. (52) and (54), we have that

$$\epsilon_s d_s - Z_i^{s(max)} < Q_i^{s(max)} \tag{54}$$

$$\Rightarrow \epsilon_s < \frac{Q_i^{s(max)} + Z_i^{s(max)}}{d_s}. \tag{55}$$

This contradicts with the given fact that $\epsilon_s = \frac{Q_i^{s(max)} + Z_i^{s(max)}}{d_s}$. Hence, we have proved that each job of type $s \in [1, S]$ is either scheduled or dropped with Alg. 1 before its maximum response delay d_s, if we set $\epsilon_s = \frac{Q_i^{s(max)} + Z_i^{s(max)}}{d_s}$.

Appendix I
Proof to Theorem 6

Since the system status, *i.e.*, the job arrival $r_i^s(t)$ and service pricing $p_i^s(t)$ of each type $s \in [1, S]$ of services and the operational price $\beta_i(t)$ at each cloud $i \in [1, F]$, changes with ergodic processes, we have that there exists a stationary randomized algorithm [19], which dynamically decides the VM valuation & pricing (with $b_i^{m*}(t)$, $\gamma_i^{m*}(t)$, $s_i^{m*}(t)$, $\eta_i^{m*}(t)$), job scheduling (with $\mu_{ij}^{s*}(t)$) & dropping (with $D_i^{s*}(t)$) and server provisioning (with $n_i^{m*}(t)$) at each cloud i, such that the offline optimum of cloud i's individual profit Ω_i^* can be achieved, together with $\bar{r}_i^s \le \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} + \bar{D}_i^{s*}$ and $\epsilon_s \le \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} + \bar{D}_i^{s*}$. Here, $\bar{a}$ denotes the time averaged expectation of variable $a(t)$.

Based on the derivations of the optimization problem (18) in and its solution in Sec. IV, we know that Algorithm 1 minimizes the right-hand-side of the drift-plus-penalty (drift-minus-utility) inequality in Eqn. (17) at each slot t, with individual profit maximization as the utility, over all possible algorithms. Then, we can have that

$$\begin{aligned}
&\Delta(\Theta_i(t)) - V \cdot [\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)] \\
&- \sum_{s\in[1,S]} D_i^s(t)\xi_i^s + \sum_{s\in[1,S]} r_i^s(t)p_i^s(t)] \\
\le& B_i + \sum_{s\in[1,S]} [Q_i^s(t)r_i^s(t) + Z_i^s(t)\epsilon_s] + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)] \\
&- V \sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) - \beta_i(t)n_i^{m*}(t)] \\
&- \sum_{s=\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^{s*}(t)[Q_i^s(t) + Z_i^s(t)] \\
&- \sum_{s\in[1,S]} D_i^{s*}(t)[Q_i^s(t) + Z_i^s(t) - V \cdot \xi_i^s] \\
=& B_i - V \cdot [\sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) - \beta_i(t)n_i^{m*}(t)] \\
&- \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s + \sum_{s\in[1,S]} r_i^s(t)p_i^s(t)] \\
&+ \sum_{s\in[1,S]} [Q_i^s(t)[r_i^s(t) - \sum_{j\in[1,F]} \mu_{ij}^{s*}(t) - D_i^{s*}(t)] \\
&+ Z_i^s(t)[\epsilon_s - \sum_{j\in[1,F]} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]].
\end{aligned}$$

Taking conditional expectations over queue status $\Theta_i(t)$ on both sides the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{\Delta(\Theta_i(t))|\Theta_i(t)\} - V \cdot \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) \\
&- \beta_i(t)n_i^m(t)] - \sum_{s\in[1,S]} D_i^s(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]|\Theta_i(t)\} \\
\le& B_i - V \cdot \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) - \beta_i(t)n_i^{m*}(t)] \\
&- \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]|\Theta_i(t)\} \\
&+ \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]|\Theta_i(t)\} \\
&+ Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]|\Theta_i(t)\}].
\end{aligned}$$

Next, we take expectations on both sides the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{\Delta(\Theta_i(t))\} - V \cdot \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) \\
&- \beta_i(t)n_i^m(t)] - \sum_{s\in[1,S]} D_i^s(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
\le& B_i - V \cdot \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) - \beta_i(t)n_i^{m*}(t)] \\
&- \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
&+ \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&+ Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

By summing over the T slots on both sides of the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{L(\Theta_i(T))\} - \mathbb{E}\{L(\Theta_i(0))\} - V \cdot \sum_{t=0}^{T-1} \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) \\
&- \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)] - \sum_{s\in[1,S]} D_i^s(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
\le& T \cdot B_i - V \cdot \sum_{t=0}^{T-1} \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) \\
&- \beta_i(t)n_i^{m*}(t)] - \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
&+ \sum_{t=0}^{T-1} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&+ Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

Since $\mathbb{E}\{L(\Theta_i(T))\} \geq 0$ and $\mathbb{E}\{L(\Theta_i(0))\} = 0$ according to the definition of the Lyapunov function, we have that

$$\begin{aligned}
&- V \cdot \sum_{t=0}^{T-1} \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^m(t)\hat{\eta}_i^m(t) - \hat{b}_i^m(t)\hat{\gamma}_i^m(t) - \beta_i(t)n_i^m(t)] \\
&- \sum_{s\in[1,S]} D_i^s(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
\le& T \cdot B_i - V \cdot \sum_{t=0}^{T-1} \mathbb{E}\{[\sum_{m\in[1,M]} [\hat{s}_i^{m*}(t)\hat{\eta}_i^{m*}(t) - \hat{b}_i^{m*}(t)\hat{\gamma}_i^{m*}(t) \\
&- \beta_i(t)n_i^{m*}(t)] - \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s + \sum_{s\in[1,S]} [r_i^s(t)p_i^s(t)]]\} \\
&+ \sum_{t=0}^{T-1} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&+ Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

Dividing $T \cdot V$ on both sides of the above inequality and taking limitation on T to infinity, we have that

$$\begin{aligned}
-\Omega_i \le& B_i/V - \Omega_i^* + \sum_{s\in[1,S]} [\bar{Q}_i^s[\bar{r}_i^s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*}] \\
&+ \bar{Z}_i^s[\epsilon_s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*}]] \\
\le& B_i/V - \Omega_i^*.
\end{aligned}$$

The second inequality comes from the fact that $\bar{r}_i^s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*} \le 0$ and $\epsilon_s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*} \le 0$. Rearranging the two sides, we have that

$$\Omega_i \geq \Omega_i^* - B_i/V,$$

APPENDIX J
PROOF TO THEOREM 7

Similar with the proof to the optimality of individual profit with Alg. 1, we have the following proof to the optimality in social welfare for our benchmark algorithm. Since the system status, *i.e.*, the job arrival $r_i^s(t)$ and service pricing $p_i^s(t)$ of each type $s \in [1, S]$ of services and the operational price $\beta_i(t)$ at each cloud $i \in [1, F]$, changes with ergodic processes, we have that there exists a stationary randomized algorithm

[19], which dynamically decides the job scheduling (with $\mu_{ij}^{s*}(t)$) & dropping (with $D_i^{s*}(t)$) and server provisioning (with $n_i^{m*}(t)$) at each cloud i, such that the offline optimum of the federation's social welfare $\prod^*$ can be achieved, together with $\bar{r}_i^{s*} \leq \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} + \bar{D}_i^{s*}$ and $\epsilon_s \leq \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} + \bar{D}_i^{s*}$. Here, $\bar{a}$ denotes the time averaged expectation of variable $a(t)$.

Based on the above derivations of the optimization problem (40) and its solution in Alg. 2, we know that Algorithm 2 minimizes the right-hand-side of the drift-plus-penalty (drift-minus-welfare) inequality in Eqn. (37) at each slot t, with social welfare maximization as the utility, over all possible algorithms. Then, we can have that

$$\begin{aligned}
&\Delta(\Theta(t)) + V \cdot \sum_{i\in[1,F]} [\sum_{m\in[1,M]} [\beta_i(t) n_i^m(t)] + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)] \\
\leq& B + \sum_{i\in[1,F]} [\sum_{s\in[1,S]} [Q_i^s(t) r_i^s(t) + Z_i^s(t)\epsilon_s] - V \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)] \\
&\quad + V \sum_{i\in[1,F]} \sum_{m\in[1,M]} [\beta_i(t) n_i^{m*}(t)] \\
&\quad - \sum_{i\in[1,F]} \sum_{s=\in[1,S]} \sum_{j\in[1,F]} \mu_{ij}^{s*}(t)[Q_i^s(t) + Z_i^s(t)] \\
&\quad - \sum_{i\in[1,F]} \sum_{s\in[1,S]} D_i^{s*}(t)[Q_i^s(t) + Z_i^s(t) - V\cdot \xi_i^s] \\
=& B + V \cdot \sum_{i\in[1,F]} [\sum_{m\in[1,M]} [\beta_i(t) n_i^{m*}(t)] + \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)] \\
&\quad + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)] \\
&\quad + Z_i^s(t)[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]].
\end{aligned}$$

Taking conditional expectations over queue status $\Theta(t)$ on both sides the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{\Delta(\Theta(t))|\Theta(t)\} + V \cdot \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^m(t) \\
&\quad + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]|\Theta(t)\} \\
\leq& B + V \cdot \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^{m*}(t) + \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]|\Theta(t)\} \\
&\quad + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]|\Theta(t)\} \\
&\quad + Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]|\Theta(t)\}].
\end{aligned}$$

Next, we take expectations on both sides the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{\Delta(\Theta(t))\} + V \cdot \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^m(t) + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
\leq& B + V \cdot \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^{m*}(t) + \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
&\quad + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&\quad + Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

By summing over the T slots on both sides of the inequality, we have that

$$\begin{aligned}
&\mathbb{E}\{L(\Theta(T))\} - \mathbb{E}\{L(\Theta(0))\} - V \cdot \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^m(t) \\
&\quad + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
\leq& T\cdot B + V \cdot \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^{m*}(t) + \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
&\quad + \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&\quad + Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

Since $\mathbb{E}\{L(\Theta(T))\} \geq 0$ and $\mathbb{E}\{L(\Theta(0))\} = 0$ according to the definition of the Lyapunov function, we have that

$$\begin{aligned}
&- V \cdot \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^m(t) + \sum_{s\in[1,S]} D_i^s(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
\leq& T\cdot B + V \cdot \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \mathbb{E}\{[\sum_{m\in[1,M]} \beta_i(t) n_i^{m*}(t) + \sum_{s\in[1,S]} D_i^{s*}(t)\xi_i^s \\
&\quad - \sum_{s\in[1,S]} r_i^s(t) p_i^s(t)]\} \\
&\quad + \sum_{t=0}^{T-1} \sum_{i\in[1,F]} \sum_{s\in[1,S]} [Q_i^s(t)\mathbb{E}\{[r_i^s(t) - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\} \\
&\quad + Z_i^s(t)\mathbb{E}\{[\epsilon_s - \sum_{j=1}^{F} \mu_{ij}^{s*}(t) - D_i^{s*}(t)]\}].
\end{aligned}$$

Dividing $T \cdot V$ on both sides and taking limitation on T to infinity, we have that

$$\begin{aligned} -\Pi \leq & B/V - \Pi^* \\ & + \sum_{i\in[1,F]} \sum_{s\in[1,S]} [\bar{Q}_i^s[\bar{r}_i^s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*}] \\ & + \bar{Z}_i^s[\epsilon_s - \sum_{j=1}^{F} \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*}]] \\ \leq & B/V - \Pi^*. \end{aligned}$$

The second inequality is based on the fact that $\bar{r}_i^s - \sum_{j=1}^F \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*} \leq 0$ and $\epsilon_s - \sum_{j=1}^F \bar{\mu}_{ij}^{s*} - \bar{D}_i^{s*} \leq 0$. Rearranging the two sides, we have that

$$\Pi \geq \Pi^* - B/V,$$

Appendix K
Proof to Theorem 8

We have shown that, Algorithm 2 achieves a social welfare with a constant gap to the offline optimum, by minimizing the RHS of *drift-plus-penalty* inequality in Eqn. (37) according to the Lyapunov optimization theory [19]. Hence, if we can prove that, Algorithm 1 can also minimize the RHS of Eqn. (37), *i.e.*, maximizing problem (38) and (39), we can also prove its social welfare optimality. Our intuition of the proof is that, when the number of clouds in the federation grows to infinity, the gap to the minimum of RHS of Eqn. (37), *i.e.*, the gap to the maximum of problem (38) and (39), by Algorithm 1 is infinitely close to zero.

– *Gap to the minimum of RHS of Eqn. (37) with Algorithm 1*: As discussed above, problem (39) is only controlled by the job drop decisions, *i.e.*, $D_i^s(t)$. Since Algorithm 1 and Algorithm 2 have the same decision on job dropping as in Eqn. (30) and Eqn. (46), the maximum of problem (39) is also achieved by Algorithm 1. Hence, the gap to the minimum of RHS of Eqn. (37), by Algorithm 1 only depends on its gap to the maximum of problem (38), which is determined by the job scheduling and server provisioning decisions.

We first map the job scheduling and server provisioning decisions in Algorithm 2 to an equivalent VM allocation based on an idealized double auction scenario. Let each cloud still proposes its buy-bid and sell-bid based on the true valuations given in Eqn. (21), (22), (25) and (26).

With Eqn. (21), we have that the price of buy-bid for VM type m at cloud i is $\frac{1}{V}$ of the maximum weight among all jobs at this cloud demanding type-m VMs. With the winner determination of our double auction mechanism, $\theta_1^m(t)$ is the maximum price of buy-bids from all clouds for type-m VMs. Hence, the cloud with buy-bid $\theta_1^m(t)$ has the maximum weight among all jobs demanding type-m VMs at all clouds.

According to the definition of $< \acute{i}_m, \acute{s}_m >$ in Eqn. (45), we know that jobs of service type $\acute{s}_m$ at cloud $\acute{i}_m$ has the maximum weight for VM type m over all service types at each cloud demanding for the same VMs. Hence, cloud $\acute{i}_m$ proposes the maximum buy-bid price $\theta_1^m(t)$, which is $\frac{1}{V}$ of the weight of type-$\acute{s}_m$ jobs at cloud $\acute{i}_m$.

With Eqn. (22), we have that the price of sell-bid for VM type m at cloud i is the larger one between i) $\frac{1}{V}$ of the maximum weight among all jobs at this cloud demanding type-m VMs; and ii) the per-VM operational price $\beta_i(t)/C_i^m$ at cloud i. With the winner determination of our double auction mechanism, $\vartheta_j^m(t)$ is the j^{th} lowest price of sell-bids from all clouds for type-m VMs.

If the maximum buy-bid price $\theta_1^m(t)$ is larger than the j^{th} lowest sell-bid price $\vartheta_j^m(t)$, we have that i) with an idealized double auction, cloud $\acute{i}_m$ has a higher buy-bid price than the sell-bid such that it can buy all VMs of type m from the cloud proposing $\vartheta_j^m(t)$; ii) with the job scheduling decision in Eqn. (44), all VMs of type m are allocated for job scheduling at cloud $\acute{i}_m$ for service type $\acute{s}_m$. Hence, the job scheduling decision in Algorithm 2 is equivalent to the idealized double auction that, the bidder with highest buy-bid price can buy all VMs from those sellers with a lower sell-bid price. Fig. 5 gives an illustration for the case. In Fig. 5, the buy-bids are sorted in descending order while the sell-bids are sorted in ascending order. Let there are k sell-bids with lower price than that of the highest buy-bid $\theta_1^m(t)$. Here, k is the maximum number of sell-bids, whose prices $\vartheta_j^m(t)$ ($j \in [1,k]$) are lower than that of the highest buy-bid $\theta_1^m(t)$, *i.e.*,

$$k = \text{arg} \max_{i\in[1,F]} \{\vartheta_j^m(t) < \theta_1^m(t) | \forall j \in [1,i]\}.$$

We know that all the k sellers sell all VMs of type m to the cloud with $\theta_1^m(t)$, *i.e.*, cloud $\acute{i}_m$.

However, the idealized cloud cannot give truthfulness guarantee. With our double auction mechanism, in Fig. 5, only the $(j'-1)$ sellers, whose sell-bid prices are no larger than the second highest buy-bid price $\theta_2^m(t)$, will sell their VMs to cloud $\acute{i}_m$. Here, j' is the maximum number of sellers with sell-bid prices no larger than $\theta_2^m(t)$, as defined in Eqn. (31). Hence, the gap to the maximum of problem (38) by Algorithm 1 is determined by the VMs that are not sold to cloud $\acute{i}_m$ by sellers between j' and k.

If these VMs of type m are allocated to cloud $\acute{i}_m$ with Algorithm 2, their utility gain in problem (38) is that

$$\begin{aligned} & \sum_{j=[j',k]} [\frac{Q_{\acute{i}_m}^{\acute{s}_m}(t) + Z_{\acute{i}_m}^{\acute{s}_m}(t)}{g_{\acute{s}_m}} - V\beta_j/C_j^m] \cdot C_j^m N_j^m \\ = & \sum_{j=[j',k]} [\theta_1^m(t) - \beta_j/C_j^m] \cdot V \cdot L_j^m(t). \end{aligned}$$

Here, the server provisioning decisions, *i.e.*, $n_i^m(t)$, are replaced by job scheduling decisions, *i.e.*, $\mu_{ij}^s(t)$, according to Eqn. (43) in Algorithm 2.

If these VMs of type m of sellers $j \in [j', k]$ are not traded to cloud $\acute{i}_m$ based on our double auction mechanism, these VMs are either scheduled to serve the jobs with maximum weight at its own cloud, if the maximum weight is higher than the per-VM operational price, or inactivated otherwise, according to the job scheduling decision in Eqn. (28) of Algorithm 1. Hence, the utility gain of these VMs for problem (38) with Algorithm 1 is that

$$\sum_{j=[j',k]} [\vartheta_j^m(t) - \beta_j/C_j^m] \cdot V \cdot L_j^m(t).$$

So, the gap to the maximum of problem (38) by Algorithm 1 is the difference between the above utility gains, as follows,

$$V\cdot\sum_{j=[j',k]}[\theta_1^m(t)-\vartheta_j^m(t)]\cdot L_j^m(t), \tag{56}$$

which is equivalent to the size of the shadow area in Fig. 5, multiplied by V.

Since $\vartheta_j^m(t)>\theta_2^m(t)$ for each $j\in[j'+1,k]$, we can give an upperbound to the gap as follows,

$$\begin{aligned}&V\cdot\sum_{j=[j',k]}[\theta_1^m(t)-\vartheta_j^m(t)]\cdot L_j^m(t),\\ =&V\cdot[[\theta_1^m(t)-\vartheta_{j'}^m(t)]\cdot L_{j'}^m(t)+\sum_{j=[j'+1,k]}[\theta_1^m(t)-\vartheta_j^m(t)]\cdot L_j^m(t)],\\ \le&V\cdot[[\theta_1^m(t)-\theta_2^m(t)+\vartheta_{j'+1}^m(t)-\vartheta_{j'}^m(t)]\cdot L_{j'}^m(t)\\ &+\sum_{j=[j'+1,k]}[\theta_1^m(t)-\theta_2^m(t)]\cdot L_j^m(t)]\\ =&V\cdot[[\vartheta_{j'+1}^m(t)-\vartheta_{j'}^m(t)]\cdot L_{j'}^m(t)+\sum_{j=[j',k]}[\theta_1^m(t)-\theta_2^m(t)]\cdot L_j^m(t)].\end{aligned} \tag{57}$$

Since the system is homogenous with the same number of servers for each VM type m at each cloud, the value of $L_j^m(t)$ at each cloud $j\in[1,F]$ is also the same. We use L^m to denote that value. Hence, the gap in Eqn. (57) can be rewritten as

$$V\cdot L^m[[\vartheta_{j'+1}^m(t)-\vartheta_{j'}^m(t)]+[k-j'][\theta_1^m(t)-\theta_2^m(t)]]. \tag{58}$$

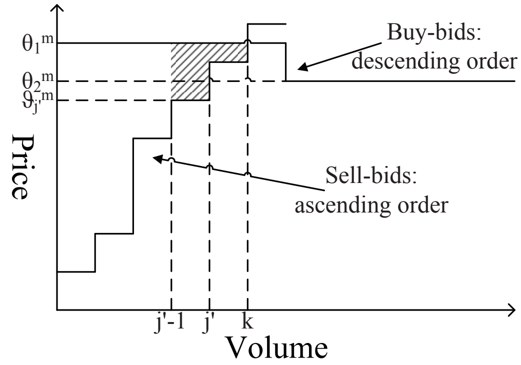


Fig. 5. Illustration of the auction.

We next show that, when the number of clouds scales to infinity, the gap in Eqn. (58) is infinitely close to zero, by analyzing the distribution of bidding prices based on the cloud number F.

– *Distribution of bidding prices*:

The distribution of the bidding prices should be analyzed in order to find an analytical result between the size of the gap in Eqn. (58) and the number of clouds F.

Let $\Theta_i(t)$ be the status of cloud i at time slot t. We see that the status $\Theta_i(t)$ is a Markov chain with Algorithm 1. Define $\mathcal{S}=\{\rho:\Pr(\Theta_i(t)=\rho|\Theta_i(0)=0)$ for some t$\}$, then $\Theta_i(t)$ is an irreducible Markov chain on state space $\mathcal{S}$ with $\Theta_i(0)=0$. This claim is true because i) any state in $\mathcal{S}$ is reachable from 0 and ii) since $\Pr(r_i^s(t)=0)>0,\ \forall s\in[1,S]$, the Markov chain can move from $\Theta_i(t)$ to 0 in finite time with a positive probability. ($Q_i^s(t)$ can be cleared by job scheduling or dropping while virtual queue $Z_i^s(t)$ can also be cleared after $Q_i^s(t)$ is zero for a constant time, according to the queueing laws Eqn. (2) and (4)) Based on the same reason as above, the state 0 is an aperiodic state. With Lemma 1 we know that $Q_i^s(t)$ and $Z_i^s(t)$ have finite upperbounds, we can then have that the state space $\mathcal{S}$ is also finite. In conclusion, the Markov chain is irreducible with an aperiodic state and finite state space. Hence, the Markov chain is ergodic.

We know that the prices of buy-bid and sell-bid for each VM type m at cloud i are calculated with its current status $\Theta_i(t)$, as well as the current operational price $\beta_i(t)$ (only for sell-bid), according Alg. 1. Since the cloud's status follows an ergodic process and the operational price is also ergodic according to our problem model, we know that the prices of buy-bid and sell-bid are also ergodic. Recall that we are proving the asymptotic optimality in social welfare by Algorithm 1 under homogenous system settings. Hence, we can have that the prices of buy-bids and sell-bids at different clouds follow the same ergodic process.

Let the price of buy-bid for VM type m follow a distribution $\mathcal{E}$ with continuous density e on the compact interval $[0,\bar{b}]$. Here, $\bar{b}=\frac{Q_i^{\bar{s}(max)}+Z_i^{\bar{s}(max)}}{V\cdot g_{\bar{s}}}$, where $\bar{s}$ is the service type with maximum value of $\frac{Q_i^{s(max)}+Z_i^{s(max)}}{V\cdot g_s}$ among all types in $[1,S]$ with $s_m=m$. Let the price of sell-bid for VM type m follow a distribution $\mathcal{H}$ with continuous density h on the compact interval $[\underline{b},\bar{b}]$. Here, $\underline{b}=\beta_i^{(min)}/C_i^m$. Denote the maximum and minimum of e and h as follows,

$$e^{min}=\min_{x\in[0,\bar{b}]}e(x)>0,\ e^{max}=\max_{x\in[0,\bar{b}]}e(x)>0,$$
$$h^{min}=\min_{x\in[\underline{b},\bar{b}]}h(x)>0,\ h^{max}=\max_{x\in[\underline{b},\bar{b}]}h(x)>0.$$

Then, we can have the following lemma according to Lemma 1 in [23],

Lemma 4:

$$\begin{aligned}\frac{1}{e^{max}(F+1)}\le&\mathbb{E}\{\theta_j-\theta_{j+1}\}\le\frac{1}{e^{min}(F+1)},\ \forall j\in[1,F-1],\\ \frac{1}{h^{max}(F+1)}\le&\mathbb{E}\{\vartheta_{j+1}-\vartheta_j\}\le\frac{1}{h^{min}(F+1)},\ \forall j\in[1,F-1].\end{aligned}$$

Here, $\mathbb{E}\{\cdot\}$ denotes the expectation.

– *Asymptotic Optimality in Minimizing the RHS of Eqn. (37) with Algorithm 1*:

With Lemma 4, we can further bound the gap in Eqn. (58) to the maximum of problem (38) by Algorithm 1 as follows,

$$\begin{aligned}&V\cdot L^m[[\vartheta_{j'+1}^m(t)-\vartheta_{j'}^m(t)]+[k-j'][\theta_1^m(t)-\theta_2^m(t)]]\\ \le&V\cdot L^m[\frac{1}{h^{min}(F+1)}+\frac{k-j'}{e^{min}(F+1)}]\end{aligned}$$

Since each seller between j' and k has a sell-bid price between $\theta_1^m(t)$ and $\theta_2^m(t)$, the sell-bid prices of all these sellers resides in an interval which has an expectation no larger than $\frac{1}{e^{min}(F+1)}$. On the other hand, since the expected interval between each sell-bid is no smaller than $\frac{1}{h^{max}(F+1)}$, we know that the expected interval between the j' seller and the k seller should be no smaller than $\frac{k-j'}{h^{max}(F+1)}$, which should be still no larger than $\frac{1}{e^{min}(F+1)}$. Hence, we have that

$$\begin{aligned}&\frac{k-j'}{h^{max}(F+1)}\le\frac{1}{e^{min}(F+1)}\\ \Rightarrow&k-j'\le\frac{h^{max}}{e^{min}}.\end{aligned}$$

Finally, we can bound the gap to the maximum of problem (38) by Algorithm 1 as follows,

$$V \cdot L^m [\frac{1}{h^{min}(F+1)} + \frac{h^{max}/e^{min}}{e^{min}(F+1)}].$$

It is clear that this gap is infinitely close to zero when the number of cloud in the federation F grows to infinity.

– *Asymptotic Optimality in Social Welfare of Algorithm 1*:

When the number of clouds in the federation scales to infinite large, *i.e.*, $F \to \infty$, the RHS of Eqn. (37) is minimized in each time slot t with our Algorithm 1 and our double auction mechanism. Thus, following the same steps, as above, to prove the social welfare optimality of Algorithm 2, which also minimizes the RHS of Eqn. (37) in each time slot t, we can prove that the time-averaged social welfare achieved with our Algorithm 1 is within a constant gap B/V from the offline optimum Π^*, when $F \to \infty$, *i.e.*,

$$\Pi \geq \Pi^* - B/V,$$